A logical approach to locality in pictures languages

Etienne Grandjean, Frédéric Olive

A logical approach to locality in pictures languages

Etienne Grandjean[a], Frédéric Olive[b]

[a] *Université de Caen Basse-Normandie / ENSICAEN / CNRS - GREYC - Caen, France*
[b] *Aix-Marseille Université / CNRS - LIF - Marseille, France*

Abstract

This paper deals with descriptive complexity of picture languages of any dimension by syntactical fragments of existential second-order logic. Two classical classes of picture languages are studied:

- The class of *recognizable* picture languages, i.e. projections of languages defined by local constraints (or tilings): it is known as the most robust class extending the class of regular languages to any dimension;

- The class of picture languages recognized on *nondeterministic cellular automata in linear time* : cellular automata is the simplest and most natural model of parallel computation and linear time is their minimal time class allowing synchronization.

We uniformly generalize to any dimension the characterization by Giammarresi et al. ("Monadic Second-Order Logic over Rectangular Pictures and Recognizability by Tiling Systems", *Inf. and Comput.* 125(1): 32–45, 1996) of the class of *recognizable* picture languages in existential monadic second-order logic.

We state several logical characterizations of the class of picture languages recognized in linear time on nondeterministic cellular automata. They are the first machine-independent characterizations of complexity classes of cellular automata.

Our characterizations are essentially deduced from normalization results we prove for first-order and existential second-order logics over pictures. They are obtained in a general and uniform framework that allows to extend them to other "regular" structures. These results show that in some sense the logics involved can be made "local" with respect to the underlying regular structures.

Finally, we describe some hierarchy results that show the optimality of our logical characterizations and delineate their limits.

Keywords: Picture languages, locality and tiling, recognizability, linear time, cellular automata, logical characterizations, monadic second-order logic, existential second-order logic.

1998 ACM Subject Classification: F.1.1 Models of Computation, F.1.3 Complexity Measures and Classes, F.4.3 Formal Languages.

Contents

Email addresses: `etienne.grandjean@unicaen.fr` (Etienne Grandjean), `frederic.olive@lif.univ-mrs.fr` (Frédéric Olive)

1. Introduction: context and discussion

Locality is a useful and widespread concept common to many areas of science: physics, chemistry, mathematics, etc. In computer science, it is a unifying notion, connecting combinatorics, logic, formal language theory, computational models, and complexity theory. For example, the local and combinatorial notion of *tiling* allowed Hao Wang and al. to prove in 1962 the undecidability of the decision problem of some logics [36, 74, 75, 3]. Locality is also a reference notion in computational complexity (e.g., see [72, 73]) and in formal language theory with the notion of *regular* or *recognizable* language that has been extended to tree or graph languages (see [68, 6]). Typically, as recalled by Borchert [2], Mac Naughton and Papert established in their classical monograph [52] that a word language is regular "iff it consists of the words whose positions can be colored so that the coloring respects the letters and obeys a given finite set of neighborhood constraints".

There is a wealth of notions of locality in *logic* and *finite model theory*. For first-order logic, Libkin's book [45] (see Chapters 4 and 5) identifies *Hanf locality* [32] and *Gaifman locality* [19] and describes a series of locality results for this logic [12, 16, 61, 33, 62, 44] and its order-invariant extension [31] or counting extension [43].

As a striking result, Gaifman's Theorem states in 1982 [19] that any first-order sentence is equivalent to a boolean combination of *local* sentences: roughly, a local sentence states the existence of k elements $x_1,\dots,x_k$, at distance $2d$ from each other (for some fixed d) such that for each x_i, the restriction of the structure to the set of elements at distance d of x_i has some fixed property ψ.

When applied to a class of structures of *bounded degree*, *e.g.* the class of cubic graphs, the local feature of first-order sentences can be even strengthened. As shown in [8, 46], such a sentence is essentially equivalent to a boolean combination of cardinality formulas with only *one variable*, i.e. of the form $\exists^k x\,\psi(x)$, meaning "there exists k elements x that satisfy $\psi(x)$".

An even stronger notion of locality in logic is presented by Borchert in [2]. There, a picture language is *local* if it is defined as the set of pictures that do not contain any pattern belonging to some fixed finite set. Borchert proves that a picture language is local iff it is definable by a first-order sentence with only *one variable* that is universally quantified, provided each picture is represented on its pixel domain with successor functions that encode the pixel adjacencies.

Computational models and *computational complexity* also involve several locality notions. Whereas it is questionable whether the Random Access Machine (RAM) or the pointer machine (e.g., see [59]) are local models, Turing machines and cellular automata are regarded as the prototypical models of *local* sequential and local parallel *computation*, respectively. Notice the *role of the underlying structure* for deciding what is local and what is nonlocal: while a configuration of a Turing machine or of a cellular automaton is essentially a word or a picture, that are *local structures*, a configuration of a RAM (resp. pointer machine) is a *function* from addresses to register contents (resp. from locations to locations). Clearly, such a function f allows to access in one step any location b from any other one a, even it they are *arbitrarily far* from each other, provided that $f(a) = b$: this contradicts the locality principle.

This paper[1] deals with *locality* in the context of *words* and *pictures* as underlying structures. For any dimension $d \geq 1$, a d-picture language is a set of d-dimensional words (colored d-dimensional grids). We study *descriptive complexity* of *nondeterministic* classes of word/picture languages by syntactical fragments of *existential second-order logic*. First, notice the following results:

1. In a series of papers culminating in [23], Giammarresi et al. proved that a 2-picture language is *recognizable* (i.e. is the projection of a local picture language) iff it is definable in *existential monadic logic* (EMSO). In short: $\mathrm{REC}^2 = \mathrm{EMSO}$. This is a picture language variant of the classical characterization of the regular/recognizable word language by (existential) monadic second-order logic, in short $\mathrm{REG} = \mathrm{REC}^1 = \mathrm{EMSO} = \mathrm{MSO}$ [4, 10, 71, 52].
2. In fact, the class REC^2 contains some NP-complete problems. More generally, one observes that for each dimension $d \geq 1$, REC^d can be defined as the class of d-picture languages recognized in *constant time* by nondeterministic d-dimensional cellular automata. That means, for each $L \in \mathrm{REC}^d$ there is some constant integer c such that each computation stops at instant c and a picture belongs to L iff it has at least one computation that stops *with each cell in an accepting state* (see e.g. [66]).

The present paper originates from two questions about word/picture languages:

- How can we generalize the proof of the above-mentioned theorem of Giammarresi et al. to any dimension? That is, can we establish the equality $\mathrm{REC}^d = \mathrm{EMSO}$ for d-picture languages of any dimension $d \geq 1$?
- Can we obtain logical characterizations of time complexity classes of cellular automata ? This originates from a question J. Mazoyer asked the first author in 2001 (personal communication): exhibit a logical characterization of the linear time complexity class of nondeterministic cellular automata.

As Cris Moore has pointed to us (personal communication), it is significant that those picture language classes – recognizable languages and picture languages recognized by time bounded cellular automata – were invented *independently* in the physics literature (see [47] for a survey). It is also well-known that those two kinds of picture languages are strongly related: it is a folklore result that the set of *time-space diagrams* of any time-bounded (nondeterministic) *cellular automaton* is a *recognizable* picture language.

A d-picture language is a set of d-pictures $p : [1,n]^d \to \Sigma$. There are two natural manners to represent such an object as a first-order structure, both presented in the literature:

[1] A preliminary and much shorter version of this paper has appeared as a conference paper [29].

- as a *pixel structure* (see e.g. [23, 22, 49, 2]): on the *pixel* domain $[1,n]^d$ where the sets $p^{-1}(s)$, $s \in \Sigma$, are encoded by unary relations $(Q_s)_{s\in\Sigma}$ and the underlying d-dimensional grid is encoded by d successor functions (see Definition 2.2);
- as a *coordinate structure* (implicitly defined and used in [2], Lemma 9.2(d)): on the *coordinate* domain $[1,n]$ where the sets $p^{-1}(s)$ are encoded by d-ary relations $(R_s)_{s\in\Sigma}$. Moreover, one uses the natural linear order of the coordinate domain $[1,n]$ and its associate successor function (see Definition 2.3).

1.1. Our main results

We establish two kinds of logical characterizations of d-picture languages, for all dimensions $d \geq 1$:

1. *On pixel structures:* $\mathrm{REC}^d = \mathrm{ESO}(\text{arity } 1) = \mathrm{ESO}(\text{var } 1) = \mathrm{ESO}(\forall^1, \text{arity } 1)$. That means a d-picture language is recognizable iff it is definable in *monadic* ESO (resp. in ESO with *one* first-order variable, or in monadic ESO with *one* universally quantified first-order variable).
2. *On coordinate structures:* $\mathrm{NLIN}^d_{\mathrm{ca}} = \mathrm{ESO}(\text{var } d+1) = \mathrm{ESO}(\forall^{d+1}, \text{arity } d+1)$; that means a d-picture language is recognized by a nondeterministic d-dimensional cellular automaton in *linear time* (see *e.g.* [67, 38]) iff it is definable in ESO with $d+1$ distinct first-order variables (resp. ESO with second-order variables of arity at most $d+1$ and a prenex first-order part of prefix $\forall^{d+1}$).

1.2. Significance of our results

Results of above Items 1 and 2 proceed from normalizations of first-order and ESO logics that we prove over picture languages. Roughly speaking, they mean that the languages of the involved complexity/logical classes are "projections" of *local* languages, or – in logical terms – are definable by ESO sentences whose first-order part is a *local formula*.

More specifically, the *normalization equality* $\mathrm{ESO}(\text{arity } 1) = \mathrm{ESO}(\forall^1, \text{arity } 1)$ of Item 1 is a consequence of the fact that, on pixel structures (and, more generally, on structures that consist of bijective functions and unary relations), any first-order formula is equivalent to a boolean combination of *cardinality formulas* of the form: "there exists k distinct elements x such that $\psi(x)$", where ψ is a quantifier-free formula with only *one* variable. The normalization of the logic – reducing it to its "one first-order variable" fragment, explicitly expresses the local feature of MSO on pictures. The results and methods of Item 1 can be summed as follows: one exploits the homogeneous framework of logic for making simpler, more explicit – using only one first-order variable – and more uniform the proof and ideas of the main result of Giammarresi et al. [23, 22]; this allows us to generalize it to *any* dimension and, potentially, to other *regular* structures.

Intuitively, our characterization $\mathrm{NLIN}^d_{\mathrm{ca}} = \mathrm{ESO}(\forall^{d+1}, \text{arity } d+1)$ of Item 2 naturally reflects a *symmetry* property of the time-space diagram of any computation of a *nondeterministic* d-dimensional cellular automaton: informally, the single first-order variable representing time *cannot be distinguished* from any of the d variables that represent the d-dimensional space. In other words, the $d+1$ variables can be permuted without increasing the expressive (or computational) power of the formula. This is the sense of the inclusion $\mathrm{ESO}(\forall^{d+1}, \text{arity } d+1) \subseteq \mathrm{NLIN}^d_{\mathrm{ca}}$ whose proof is far from trivial: roughly speaking, it needs to normalize any sentence of $\mathrm{ESO}(\forall^{d+1}, \text{arity } d+1)$ by "sorting" its $d+1$ first-order variables so that the sentence so "sorted" becomes *local*. As expected, this "sorted" $\mathrm{ESO}(\forall^{d+1}, \text{arity } d+1)$ characterization of $\mathrm{NLIN}^d_{\mathrm{ca}}$ for coordinate representation of pictures is the exact equivalent of the $\mathrm{ESO}(\forall^1, \text{arity } 1)$ characterization of recognizable picture languages in pixel representation: the set of *time-space diagrams* of a (nondeterministic) cellular automaton is *recognizable*.

Notice that Borchert [2] has stated some results to be compared with our logical characterizations of recognizable languages and of linear time bounded complexity classes of multidimensional cellular automata. However, paradoxically, his paper never mentions cellular automata. To avoid technicalities in this introduction, we will describe and discuss the results of [2], less general than ours, in Section 11.

1.3. Additional results

The last part of this paper consists of an attempt to answer two natural questions related to each other and concerning the frameworks/meaning of our main results and the possible extensions of these results:

- *Question 1: Why two distinct frameworks?* Why our logical characterizations of the class, on one hand, of recognizable pictures, i.e. EMSO on pixel encoding, and on the other hand, of the linear-time complexity class of nondeterministic d-dimensional cellular automata, i.e. $\mathrm{ESO}(\mathrm{var}\, d)$ on coordinate encoding, do *not* involve the same logical framework with the same encoding?
- *Question 2: Strict hierarchies:* Which *strict* hierarchy results can we prove among the various definability classes we have studied?

Question 1: two frameworks. First, notice that for dimension one, i.e. for word languages, the pixel representation and the coordinate one trivially coincide. So the logical characterizations

$$\mathrm{REC}^1 = \mathrm{REG} = \mathrm{ESO}(\mathrm{arity}\ 1) = \mathrm{ESO}(\mathrm{var}\ 1) = \mathrm{ESO}(\forall^1, \mathrm{arity}\ 1) \text{ and}$$
$$\mathrm{NLIN}^1_{\mathrm{ca}} = \mathrm{ESO}(\mathrm{var}\ 2) = \mathrm{ESO}(\forall^2, \mathrm{arity}\ 2)$$

are established in a unique framework for words. Let us now justify our distinct frameworks for any larger dimension $d > 1$.

- *Coordinate representation does not fit* REC^d: The class of recognizable d-picture languages *cannot* be *naturally* characterized in logic *with coordinate representation* because of the following *strict* inclusions established in Section 9.2:

$$\mathrm{ESO}(\mathrm{var}\, d-1) \subsetneq \mathrm{REC}^d \subsetneq \mathrm{ESO}(\mathrm{var}\, d).$$

Intuitively, the logic $\mathrm{ESO}(\mathrm{var}\, d)$ – or its equivalent restriction $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$ – is *not local* when it is applied to the coordinate representations of d-pictures, for $d \geq 2$. However, we will establish in Proposition 9.11 that such a locality can be obtained by "folding" the pictures: a d-picture language L is definable in $\mathrm{ESO}(\mathrm{var}\, d)$ for coordinate representation iff the set of folded versions of pictures of L belongs to REC^d.

- *Pixel representation does not fit* $\mathrm{NLIN}^d_{\mathrm{ca}}$: We will prove in Section 9.2 the following sequence of strict inclusions for pixel representation of d-pictures and $d > 1$:

$$\mathrm{ESO}(\mathrm{var}\ 1) \subsetneq \mathrm{NLIN}^d_{\mathrm{ca}} \subsetneq \mathrm{ESO}(\mathrm{var}\ 2)$$

This justifies that no logic of the form $\mathrm{ESO}(\mathrm{var}\, k)$ – or, equivalently, $\mathrm{ESO}(\forall^k, \mathrm{arity}\, k)$) –, for any k, can characterize the class $\mathrm{NLIN}^d_{\mathrm{ca}}$ for pixel representation.

Question 2: Strict hierarchies. Our final result (see Section 10) is a strict hierarchy result for d-languages represented by *coordinate* structures. It shows some subtile relationships between the number of first-order variables and the arity of the ESO relation symbols.

Theorem 10.6. *For each integer $d \geq 2$ and for d-languages represented by* coordinate *structures, the following (strict) inclusions hold:*

$$\begin{array}{ccccc} \mathrm{REC}^d & \subsetneq & \mathrm{ESO}(\mathrm{var}\, d) & = & \mathrm{ESO}(\forall^d, \mathrm{arity}\, d) \\ & & & & \subsetneq \\ & & & & \mathrm{ESO}(\forall^{d+1}, \mathrm{arity}\, d) \\ & & & & \subseteq \\ \mathrm{NLIN}^d_{ca} & = & \mathrm{ESO}(\mathrm{var}\, d+1) & = & \mathrm{ESO}(\forall^{d+1}, \mathrm{arity}\, d+1) \end{array}$$

Theorem 10.6 straightforwardly yields the following separation result:

$$\mathrm{ESO}(\mathrm{var}\, d) \subsetneq \mathrm{ESO}(\mathrm{arity}\, d)$$

for coordinate representation of d-picture languages with $d \geq 2$. This striking result contrasts with the equality $\mathrm{ESO}(\mathrm{var}\, 1) = \mathrm{ESO}(\mathrm{arity}\, 1)$ for the pixel representation. To our knowledge, this is also the first result that shows that the *arity* is *strictly* more expressive than the *number of first-order variables* in ESO definability. We should mention that the proof of this separation result and of Theorem 10.6 involves in an essential manner the hypothesis that the arity of the ESO predicates *equals* the arity of the input predicates. If the ESO arity is *greater* than the input arity then those separation problems remain open.

1.4. Structure of the paper

After Section 2 recalls succinctly some definitions – pictures, picture languages, the two encodings of pictures and logical notions –, Section 3 proves our first main result: logical characterizations of the class REC of recognizable picture languages.

The long proof of our second main result, the logical characterizations of $\mathrm{NLIN}^d_{\mathrm{ca}}$, is presented along Sections 4 to 8. A very long part of the proof consists in successively normalizing the logic $\mathrm{ESO}(\mathrm{var}\, d)$ into more and more restrictive forms, the last one being $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \mathrm{sorted})$.

A recapitulation and a comparison of all the previous results is given in Section 9 with a formal proof that the same – pixel or coordinate – representation cannot be used for characterizing both classes REC and $\mathrm{NLIN}^d_{\mathrm{ca}}$; however, we show how "folding" the pictures allows to relate in some manner the involved logics and both picture representations.

Finally, some hierarchy results are proved in Section 10 and Section 11 presents some additional results, open problems and final remarks.

2. Preliminaries

In the definitions below and all along the paper, we denote by Σ, Γ some finite alphabets and by d a positive integer. For any positive integer n, we set $[n] := \{1, \ldots, n\}$. We are interested in sets of pictures of any fixed dimension d.

Definition 2.1. *A d-dimensional picture or d-picture on Σ is a function $p : [n]^d \to \Sigma$ where n is a positive integer. The set $dom(p)$= $[n]^d$ is called the domain of picture p and its elements are called points, pixels or cells of the picture. A set of d-pictures on Σ is called a d-dimensional picture language, or d-language, on Σ.*

Notice that 1-pictures on Σ are nothing but nonempty words on Σ.

2.1. Pictures as model theoretic structures

Along the paper, we will often describe d-languages as sets of models of logical formulas. To allow this point of view, we must settle on an encoding of d-pictures as model theoretic structures.

For logical aspects of this paper, we refer to the usual definitions and notations in logic and finite model theory (see [9] [45], or [26]). A *signature* (or *vocabulary*) σ is a finite set of relation and function symbols each of which has a fixed arity. A (finite) *structure* S of vocabulary σ, or σ-structure, consists of a finite domain D of cardinality $n \geq 1$, and, for any symbol $s \in \sigma$, an interpretation of s over D, often denoted by s for simplicity. The tuple of the interpretations of the σ-symbols over D is called the *interpretation* of σ over D and, when no confusion results, it is also denoted σ. The *cardinality of a structure* is the cardinality of its domain. For any signature σ, we denote by $\textsc{struc}(\sigma)$ the class of (finite) σ-structures. We write $\textsc{models}(\Phi)$ the set of σ-structures which satisfy some fixed formula Φ. We will often deal with *tuples* of objects. We denote them by bold letters.

There are two natural manners to represent a picture by some logical structure: on the domain of its pixels, or on the domain of its coordinates. This gives rise to the following definitions:

Definition 2.2. *The* pixel structure*, or* pixel encoding*, of a picture $p : [n]^d \to \Sigma$ is the structure*

$$\mathsf{pixel}^d(p) = ([n]^d, (Q_s)_{s\in\Sigma}, (succ_i)_{i\in[d]}, (min_i)_{i\in[d]}, (max_i)_{i\in[d]}).$$

where

- *$succ_j$ is the* (cyclic) successor function *according to the j^{th} dimension of $[n]^d$, mapping each $(a_1, \dots, a_d) \in [n]^d$ on $(a_1^{(j)}, \dots, a_d^{(j)}) \in [n]^d$, where we set : $a_i^{(j)} = a_i$ for $i \neq j$ and, beside, $a_j^{(j)} = a_j + 1$ if $a_j < n$; $a_j^{(j)} = 1$ otherwise;*

  in other words, for $\mathbf{a} \in [n]^d$, $succ_j(\mathbf{a})$ is the d-tuple $\mathbf{a}^{(j)}$ obtained from $\mathbf{a}$ by "increasing" its j^{th} component according to the cyclic successor on $[n]$;

- *the min_i's, max_i's and Q_s's are the following unary (monadic) relations:*

$$min_i = \{\mathbf{a} \in [n]^d : a_i = 1\}; \qquad max_i = \{\mathbf{a} \in [n]^d : a_i = n\}; \qquad Q_s = \{\mathbf{a} \in [n]^d : p(\mathbf{a}) = s\}.$$

Definition 2.3. *The* coordinate structure*, or* coordinate encoding*, of a picture $p : [n]^d \to \Sigma$ is the structure*

$$\mathsf{coord}^d(p) = \langle [n], (Q_s)_{s\in\Sigma}, <, succ, min, max\rangle \tag{1}$$

where

- *for $s \in \Sigma$, Q_s is a d-ary relation symbol interpreted as the set of cells of p labelled by s; in other words:*

$$Q_s = \{\mathbf{a} \in [n]^d : p(\mathbf{a}) = s\};$$

- *$<$, min, max are predefined relation symbols of respective arities 2, 1, 1, that are interpreted, respectively, as the sets $\{(i, j) : 1 \leq i < j \leq n\}$, $\{1\}$ and $\{n\}$;*

- *$succ$ is a unary function symbol interpreted as the cyclic successor (that is: $succ(i) = i + 1$ for $i < n$ and $succ(n) = 1$).*

For a d-language L, we set $\mathsf{pixel}^d(L) = \{\mathsf{pixel}^d(p) : p \in L\}$ and $\mathsf{coord}^d(L) = \{\mathsf{coord}^d(p) : p \in L\}$.

Remark 2.4. *In the sequel, we often write, for some logical property P, that "P holds on pixel encodings", or "P holds on d-pixel encodings". Naturally, it means the property P holds for all structure of the form* $\mathsf{pixel}(p)$ *(or* $\mathsf{pixel}^d(p)$ *in the latter case). The same formulations occur for "coordinate encodings"*

Remark 2.5. *Several details are irrelevant in Definitions 2.2 and 2.3, i.e. our results still hold for several variants, in particular:*

- *In Definition 2.3, the fact that the linear order < and the equality = are allowed or not and the fact that min, max are represented by individual constants or unary relations;*
- *In both definitions, the fact that the successor function(s) is/are cyclic or not and is/are completed or not by predecessor(s) function(s).*

At the opposite, it is essential that, in both definitions:

- *The successor(s) is/are represented by* function(s) *and not by (binary) relation(s);*
- *The min, max are explicitly represented.*

2.2. Logics under consideration

Let us now come to the logics involved in the paper. All formulas considered hereafter belong to *relational Existential Second-Order logic*. Given a signature σ, indifferently made of relational and functional symbols, a relational existential second-order formula of signature σ has the shape $\Phi \equiv \exists \mathbf{R} \varphi(\sigma, \mathbf{R})$, where $\mathbf{R} = (R_1, \ldots, R_k)$ is a tuple of relational symbols and φ is a first-order formula of signature $\sigma \cup \{\mathbf{R}\}$. We denote by ESO^σ the class thus defined. We will often omit to mention σ for considerations on these logics that do not depend on the signature. Hence, ESO stands for the class of all formulas belonging to ESO^σ for some σ.

We will pay great attention to several variants of ESO. In particular, we will distinguish formulas of type $\Phi \equiv \exists \mathbf{R} \varphi(\sigma, \mathbf{R})$ according to: the number of distinct first-order variables involved in φ, the arity of the second-order symbols $R \in \mathbf{R}$, and the quantifier prefix of some prenex form of φ.

With the logic $\mathrm{ESO}^\sigma(\forall^d, \text{arity } \ell)$, we control these three parameters: it is made of formulas of which first-order part is prenex with a universal quantifier prefix of length d, and where existentially quantified relation symbols are of arity at most ℓ. In other words, $\mathrm{ESO}^\sigma(\forall^d, \text{arity } \ell)$ collects formulas of shape $\exists \mathbf{R} \forall \mathbf{x} \theta(\sigma, \mathbf{R}, \mathbf{x})$ where θ is quantifier free, $\mathbf{x}$ is a d-tuple of first-order variables, and $\mathbf{R}$ is a tuple of relation symbols of arity at most ℓ. Relaxing some constraints of the above definition, we set:

$$\mathrm{ESO}^\sigma(\forall^d) = \bigcup_{\ell>0} \mathrm{ESO}^\sigma(\forall^d, \text{arity } \ell) \text{ and } \mathrm{ESO}^\sigma(\text{arity } \ell) = \bigcup_{d>0} \mathrm{ESO}^\sigma(\forall^d, \text{arity } \ell).$$

Finally, we write $\mathrm{ESO}^\sigma(\text{var } d)$ for the class of formulas that involve at most d first-order variables, thus focusing on the sole number of distinct first-order variables (possibly quantified several times).

In the following sections, we'll prove that some logics have the same expressive power, as far as given sets of structures are concerned. When a normalization of a logic $\mathcal{L}$ into a logic $\mathcal{L}'$ is thus relativized to a specific class $\mathcal{S}$ of structures, we write: $\mathcal{L} = \mathcal{L}'$ on $\mathcal{S}$. The next definition details the meaning of this formulation.

Definition 2.6. *Given a set of structures* $\mathcal{S}$ *and a formula* Φ, *the set of models of* Φ *that belong to* $\mathcal{S}$ *is denoted by* $\textsc{models}_\mathcal{S}(\Phi)$. *Two formulas* Φ *and* Φ' *are* $\mathcal{S}$-equivalent *if* $\textsc{models}_\mathcal{S}(\Phi) = \textsc{models}_\mathcal{S}(\Phi')$. *Given two logics* $\mathcal{L}$ *and* $\mathcal{L}'$, *we say that* $\mathcal{L} \subseteq \mathcal{L}'$ on $\mathcal{S}$ *if each* $\Phi \in \mathcal{L}$ *is* $\mathcal{S}$*-equivalent to some* $\Phi' \in \mathcal{L}'$. *Furthermore, we write* $\mathcal{L} = \mathcal{L}'$ on $\mathcal{S}$ *if both* $\mathcal{L} \subseteq \mathcal{L}'$ *and* $\mathcal{L}' \subseteq \mathcal{L}$ *hold on* $\mathcal{S}$.

In some very rare cases, we will consider the extension of ESO obtained by allowing quantification over functional symbols. The corresponding logic, ESOF, gathers all formulas of the form $\exists \mathbf{R} \exists \mathbf{f} \varphi(\sigma, \mathbf{R}, \mathbf{f})$, where $\mathbf{R}$ (resp. $\mathbf{f}$) is a tuple of relational (resp. functional) symbols and φ is any first-order formula of signature $\sigma \cup \{\mathbf{R}, \mathbf{f}\}$. The restrictions $\mathrm{ESOF}(\mathrm{var}\, d)$ and $\mathrm{ESOF}(\forall^d, \mathrm{arity}\, \ell)$ of ESOF are defined as for ESO. The expressive power of these logics is quite high. A $\sigma-$NRAM is a nondeterministic Random Access Machine that takes σ-structures as inputs in the following way: for each $s \in \sigma$ of arity ℓ and each ℓ-tuple $\mathbf{t} \in D^\ell$, a special register $[s, \mathbf{t}]$ contains the value of $s(\mathbf{t})$. Let $\mathrm{NTIME}^\sigma(n^d)$ be the class of problems on σ-structures decidable by a $\sigma-$NRAM in time $O(n^d)$ where n is the size of the domain D of structures. The following was proved in [28]:

Theorem 2.7 ([28]). *For all $d > 0$, $\mathrm{NTIME}(n^d) = \mathrm{ESOF}(\mathrm{var}\, d)$.*

In the same paper, a normalization of $\mathrm{ESOF}(\mathrm{var}\, d)$ was stated:

Proposition 2.8 ([28]). *For all $d > 0$, $\mathrm{ESOF}(\mathrm{var}\, d) = \mathrm{ESOF}(\forall^d, \mathrm{arity}\, d)$.*

3. Recognizable picture languages and their logical characterizations

In this section, we define the class of local (resp. recognizable) picture languages and establish the logical characterizations of the class of recognizable picture languages.

3.1. Local and recognizable picture languages

Our notion of *local picture language* or *tilings language* is based on some sets of allowed patterns (called tiles) of the *bordered* pictures. It is a simple generalization to any dimension of the notion of *hv-local* 2-dimensional picture language of [41] (see also [22, 20, 21, 23, 24, 2]). To define it formally, we need to mark the border of pictures.

Definition 3.1. *By $\Gamma^\sharp$ we denote the alphabet $\Gamma \cup \{\sharp\}$ where $\sharp$ is a special symbol not in Γ. Let p be any d-picture of domain $[n]^d$ on Γ. The bordered d-picture of p, denoted by $p^\sharp$, is the function $p^\sharp : [0, n+1]^d \to \Gamma^\sharp$ defined by $p^\sharp(a) = p(a)$ if $a \in dom(p)$; $p^\sharp(a) = \sharp$ otherwise. Here, "otherwise" means that a is on the border of $p^\sharp$, that is, some component a_i of a is 0 or $n+1$.*

Here is our definition of *local picture language* or *tilings language*.

Definition 3.2.

1. *Given a d-picture p and an integer $j \in [d]$, two cells $a = (a_i)_{i\in[d]}$ and $b = (b_i)_{i\in[d]}$ of p are j-adjacent if they have the same coordinates, except the j^{th} one for which $|a_j - b_j| = 1$.*
2. *A tile for a d-language L on Γ is a pair in $(\Gamma^\sharp)^2$.*
3. *A picture $p : [n]^d \to \Gamma$ is j-tiled by a set of tiles $\Delta \subseteq (\Gamma^\sharp)^2$ if for any two j-adjacent points $a, b \in dom(p^\sharp)$: $(p^\sharp(a), p^\sharp(b)) \in \Delta$.*
4. *Given d sets of tiles $\Delta_1, \ldots, \Delta_d \subseteq (\Gamma^\sharp)^2$, a d-picture p is tiled by $(\Delta_1, \ldots, \Delta_d)$ if p is j-tiled by Δ_j for each $j \in [d]$.*
5. *We denote by $L(\Delta_1, \ldots, \Delta_d)$ the set of d-pictures on Γ that are tiled by $(\Delta_1, \ldots, \Delta_d)$.*
6. *A d-language L on Γ is local if there exist $\Delta_1, \ldots, \Delta_d \subseteq (\Gamma^\sharp)^2$ such that $L = L(\Delta_1, \ldots, \Delta_d)$. We then say that L is $(\Delta_1, \ldots, \Delta_d)$-local, or $(\Delta_1, \ldots, \Delta_d)$-tiled.*

Remark 3.3. *This notion of* locality *is more restrictive than that given by Giammarresi and al. [23]. However, this does not affect the notion of* recognizability *as defined below, a robust notion that remains equivalent to that defined in [23].*

Definition 3.4. *A d-language L on Σ is* recognizable *if it is the projection (i.e. homomorphic image) of a* local *d-language over an alphabet Γ. It means there exist a surjective function $\pi : \Gamma \to \Sigma$ and a local d-language L_{loc} on Γ such that*

$$L = \{\pi(p) : \ p \in L_{loc}\}.$$

where of course $\pi(p)$ means $\pi \circ p$. Because of the last item of Definition 3.2, one can also write: L is recognizable if there exist a surjective function $\pi : \Gamma \to \Sigma$ and d sets $\Delta_1, \ldots, \Delta_d \subseteq (\Gamma^\sharp)^2$ such that

$$L = \{\pi(p) : \ p \in L(\Delta_1, \ldots, \Delta_d)\}.$$

We denote by REC^d *the class of recognizable d-languages.*

3.2. Logical characterizations of recognizable picture languages

A characterization of recognizable languages of dimension 2 by a fragment of existential monadic second-order logic was proved by Giammarresi et al. [23]. They established:

Theorem 3.5 ([23]). *For any 2-language L: $L \in \mathrm{REC}^2 \Leftrightarrow \mathsf{pixel}^2(L) \in \mathrm{ESO}(\text{arity } 1)$.*

In this section, we return to this result. We simplify its proof, refine the logic it involves, and generalize its scope to any dimension.

Theorem 3.6. *For any $d > 0$ and any d-language L, the following assertions are equivalent:*

1. $L \in \mathrm{REC}^d$*;*
2. $\mathsf{pixel}^d(L) \in \mathrm{ESO}(\forall^1, \text{arity } 1)$*;*
3. $\mathsf{pixel}^d(L) \in \mathrm{ESO}(\text{arity } 1)$.

Theorem 3.6 is a straightforward consequence of the forthcoming Propositions 3.9 and 3.14. The former states the equivalence of Items 1 and 2 above; the latter establishes the normalization $\mathrm{ESO}(\text{arity } 1) = \mathrm{ESO}(\forall^1, \text{arity } 1)$ on pixel structures.

In order to prove Proposition 3.9, it is convenient to first normalize the sentences of $\mathrm{ESO}(\forall^1, \text{arity } 1)$. This is the role of the technical result below, which asserts that on pixel encodings, such a sentence can be rewritten in a very local form where the first-order part alludes only pairs of adjacent pixels of the bordered picture.

Lemma 3.7. *On pixel structures, any sentence $\varphi \in \mathrm{ESO}(\forall^1, \text{arity } 1)$ is equivalent to a sentence of the form:*

$$\exists \mathbf{U} \forall x \bigwedge_{i \in [d]} \left\{ \begin{array}{rcll} min_i(x) & \to & \mathsf{m}_i(x) & \wedge \\ max_i(x) & \to & \mathsf{M}_i(x) & \wedge \\ \neg max_i(x) & \to & \Psi_i(x) & \end{array} \right\}. \tag{2}$$

Here, $\mathbf{U}$ is a list of monadic relation variables and m_i, M_i, Ψ_i are quantifier-free formulas such that

- *atoms of m_i and M_i have all the form $Q(x)$,*

- *atoms of Ψ_i have all the form $Q(x)$ or $Q(succ_i(x))$,*

where, in both cases, $Q \in \{(Q_s)_{s\in\Sigma}, \mathbf{U}\}$.

Proof. Let $\varphi \in \mathrm{ESO}(\forall^1, \text{arity } 1)$ be a sentence on pixel structures.

Suppression of equalities: Without loss of generality, assume that φ is in negative normal form[2] $\exists \mathbf{U} \forall x \psi$ and that each equality in ψ is of the form

$$\mathrm{succ}_{i_1}^{c_{i_1}} \dots \mathrm{succ}_{i_k}^{c_{i_k}}(x) = x \tag{3}$$

where the k indices $i_1, \dots, i_k$ ($k > 0$) are pairwise distinct, and the exponents $c_{i_1}, \dots, c_{i_k}$ are positive integers. Equation (3) holds in some pixel structure $\mathsf{pixel}^d(p)$ of domain $[n]^d$ for some x (or, equivalently, for all x), iff the side n of p satisfies equalities $n = c_{i_1} = c_{i_2} = \dots = c_{i_k}$. So, we have to suppress any equality/inequality of the form $n = c$ or $n \neq c$, for $c \geq 1$, in φ. First, notice that the inequality $n \neq c$ can be rewritten as

$$n > c \vee \bigvee_{j\in[c-1]} n = j.$$

So, there remains to suppress such an equality/inequality $n = c$ or $n > c$, for $c \geq 1$. This can be done by introducing $c+1$ new unary relation symbols denoted $\mathrm{coord}_1^{=j}(x)$, for $j \in [c+1]$. Intuitively, $\mathrm{coord}_1^{=j}(x)$ means: "the first coordinate of point x is j". Clearly, for any $c > 0$, the unary relations $\mathrm{coord}_1^{=j}$ are defined by induction on $j \in [c+1]$ with the formula $\forall x\, \delta^c(x)$ where

$$\delta^c = (\min_1(x) \leftrightarrow \mathrm{coord}_1^{=1}(x)) \wedge \bigwedge_{j\in[c]} (\neg\max_1(x) \to (\mathrm{coord}_1^{=j}(x) \leftrightarrow \mathrm{coord}_1^{=j+1}(\mathrm{succ}_1(x)))).$$

Using those relations, it is rather easy to see that the two formulas

$$\forall x\, (\max_1(x) \to \mathrm{coord}_1^{=c}(x)) \text{ and } \forall x \bigwedge_{j\in[c]} (\mathrm{coord}_1^{=j}(x) \to \mathrm{coord}_1^{=j+1}(\mathrm{succ}_1(x)))$$

express the assertions $n = c$ and $n > c$, respectively. Hence, the first-order sentence $\forall x\, \psi(x)$ is equivalent to the $\mathrm{ESO}(\forall^1, \text{arity } 1)$-sentence:

$$\exists \mathbf{coord}\, \forall x : \ \delta(x) \wedge \psi'(x)$$

where **coord** denotes the list of unary relation variables $\mathrm{coord}_1^{=j}$ introduced in the required formulas $\delta^c(x)$, the conjunction of which is denoted $\delta(x)$, and $\psi'(x)$ is the formula $\psi(x)$ where each "sub-formula" $n = c$ (resp. $n > c$) is replaced by the equivalent formula $\max_1(x) \to \mathrm{coord}_1^{=c}(x)$ (resp. $\bigwedge_{j\in[c]} (\mathrm{coord}_1^{=j}(x) \to \mathrm{coord}_1^{=j+1}(\mathrm{succ}_1(x)))$).

So, our sentence φ can be assumed to be in prenex conjunctive normal form $\exists \mathbf{U} \forall x \psi$ *without equality*, that means ψ is a conjunction of clauses with literals of the form $Q(\tau(x))$ or $\neg Q(\tau(x))$ where Q belongs to the set of relations $\{(\min_i)_{i\in[d]}, (\max_i)_{i\in[d]}, (Q_s)_{s\in\Sigma}, \mathbf{U}\}$ and τ is a (possibly empty) composition of function symbols succ_i, $i \in [d]$. The idea is to introduce for each atom $Q(\tau(x))$ that occurs in ψ a new unary relation variable denoted $U_{Q,\tau}$ so that the atom $U_{Q,\tau}(x)$ is equivalent to (can replace) the atom $Q(\tau(x))$.

The $U_{Q,\tau}$'s are defined inductively by the conjunction of the following equivalences, denoted by *basic* and *$succ_i$-induct*:

[2] That means the scope of each negation is an atomic formula.

- *basic*: $U_{Q,Id}(x) \leftrightarrow Q(x)$,
- *succ$_i$-induct*: $U_{Q,\tau\mathrm{succ}_i}(x) \leftrightarrow U_{Q,\tau}(\mathrm{succ}_i(x))$

from which the equivalence claimed $U_{Q,\tau}(x) \leftrightarrow Q(\tau(x))$ can be deduced.

Let $\delta(x)$ denote the conjunction of all the equivalences that define the $U_{Q,\tau}$'s and let $\psi'(x)$ denote the formula $\psi(x)$ where each atom $Q(\tau(x))$ is replaced by the atom $U_{Q,\tau}(x)$. Clearly, the sentence $\varphi = \exists\mathbf{U}\forall x\psi(x)$ is equivalent to the sentence

$$\varphi' = \exists\mathbf{U}\exists(U_{Q,\tau})'_s\,\forall x\,(\psi'(x) \wedge \delta(x)).$$

Now, put φ', that means $\psi'(x) \wedge \delta(x)$, in conjunctive normal form. In order to organize and transform the clauses of φ', some terminology is required about clauses:

- a clause is *x-pure* (resp. *i-cyclic*) if it only contains atoms of the form $Q(x)$ (resp. $Q(x)$ or $Q(\mathrm{succ}_i(x))$) where Q is a unary relation symbol which is neither any $\min_j$ nor any $\max_j$;
- an *i-local* clause is of the form $\neg\max_i(x) \to C(x)$ where $C(x)$ is an i-cyclic clause;
- an *i-min* (resp. *i-max*) clause is of the form $\min_i(x) \to C(x)$ (resp. $\max_i(x) \to C(x)$) where $C(x)$ is an x-pure clause.

Using those definitions, we observe that

- the clauses of the conjunctive normal form of $\psi'(x)$ are x-pure;
- the *succ$_i$-induct* clauses of $\delta(x)$ are i-cyclic;
- each *basic* implication of $\delta(x)$ of the form $U_{Q,Id}(x) \to Q(x)$ or $Q(x) \to U_{Q,Id}(x)$ is an x-pure clause except in case Q is $\min_i$ or $\max_i$ ($i \in [d]$); clearly, in this case, the implication can be rephrased in one of the following four forms:
  1. $\min_i(x) \to C(x)$,
  2. $\max_i(x) \to C(x)$,
  3. $\neg\min_i(x) \to C(x)$,
  4. $\neg\max_i(x) \to C(x)$,

  where $C(x)$ is an x-pure clause (literal). Clauses 1 and 2 are i-min and i-max clauses, respectively. Clause 4 is i-local. Clause 3 can be replaced by the i-local clause $\neg\max_i(x) \to C(\mathrm{succ}_i(x))$; this is justified by the equivalence (easily proved) of the universally quantified versions of those implications:

  $$\forall x(\neg\min_i(x) \to C(x)) \Leftrightarrow \forall x(\neg\max_i(x) \to C(\mathrm{succ}_i(x))).$$

So, we have shown how to rephrase the first-order part (that is $\psi'(x) \wedge \delta(x)$) of φ' as a conjunction of x-pure clauses, i-cyclic clauses, i-local clauses, i-min clauses and i-max clauses. In fact, all those clauses are local with the exception of i-cyclic clauses. Recall that an i-cyclic clause $C(x, \mathrm{succ}_i(x))$ only contains atoms of the two forms $Q(x)$ or $Q(\mathrm{succ}_i(x))$) where Q is a unary relation symbol which is neither any $\min_j$ nor any $\max_j$. Its nonlocality is due to the following fact: if for a d-picture p we have $a \in \max_i$ for any pixel $a \in dom(p)$, then the pixel $\mathrm{succ}_i(a)$ is not adjacent to a in p since we have $\mathrm{succ}_i(a) \in \min_i$ by cyclicity of the function (permutation) succ_i. In order to recover locality, let us first replace each i-cyclic clause $C(x, \mathrm{succ}_i(x))$ by the equivalent conjunction of the following two clauses:

1. the i-local clause $\neg\max_i(x) \to C(x, \mathrm{succ}_i(x))$;
2. the "nonlocal" clause $\max_i(x) \to C(x, \mathrm{succ}_i(x))$.

So, there remains to get rid of the "nonlocal" clause 2. The trick consists in making available in all the points of any succ_i-cycle the value of each unary relation Q for the $\min_i$ point of this cycle. This can be done by using a new unary relation symbol $U^Q_{min,i}(x)$ defined inductively by the conjunction of the following $\min_i$ and i-local clauses

$$\min_i(x) \to (U^Q_{min,i}(x) \leftrightarrow Q(x))$$

$$\neg\max_i(x) \to (U^Q_{min,i}(x) \leftrightarrow U^Q_{min,i}(\mathrm{succ}_i(x))).$$

Clearly, for each point a of any succ_i-cycle γ, we have the constant value $U^Q_{min,i}(a) = Q(b)$ where b is the unique point in $\gamma \cap \min_i$. A new unary relation symbol $U^Q_{max,i}$ can be defined similarly for $\max_i$. This justifies that each "nonlocal" clause $\max_i(x) \to C(x, \mathrm{succ}_i(x))$ can be replaced by the x-pure clause $C'(x)$ obtained by substituting in the clause $C(x, \mathrm{succ}_i(x))$ each atom $Q(x)$ (resp. $Q(\mathrm{succ}_i(x))$) by $U^Q_{max,i}(x)$ (resp. $U^Q_{min,i}(x)$).

Let us recapitulate what we have obtained. Our initial sentence $\varphi = \exists\mathbf{U}\forall x\, \psi(x)$ of $\mathrm{ESO}(\forall^1, \text{arity } 1)$ is logically equivalent to a sentence of the form $\exists\mathbf{U}'\forall x\, \Psi(x)$ where

- $\mathbf{U}'$ is the union of the set of ESO unary symbols of φ', that are $\mathbf{U}$ and the $U_{Q,\tau}$'s, and of the $U^Q_{min,i}$'s and $U^Q_{max,i}$'s we have just introduced;
- $\Psi(x)$ is a conjunction of x-pure clauses, i-min clauses, i-max clauses and i-local clauses.

Now, it is easy to transform the conjunction of clauses $\Psi(x)$ into the conjunction of formulas required:

$$\bigwedge_{i\in[d]} [(\min_i(x) \to \Psi_i^{min}(x)) \wedge (\max_i(x) \to \Psi_i^{max}(x)) \wedge (\neg\max_i(x) \to \Psi_i(x))].$$

More precisely, for each $i \in [d]$,

- the conjunction of the i-min clauses (resp. the i-max clauses) and the x-pure clauses of $\Psi(x)$ is trivially transformed into the required form $\min_i(x) \to \Psi_i^{min}(x)$ (resp. $\max_i(x) \to \Psi_i^{max}(x)$);
- the conjunction of the i-local clauses and the x-pure clauses of $\Psi(x)$ is similarly transformed into the required form $\neg\max_i(x) \to \Psi_i(x)$.

This completes the proof of Lemma 3.7. ◻

Remark 3.8. *The normal form of the formula obtained in Lemma 3.7 guarantees its local feature. In particular, notice that any successor symbol succ_i can only apply to arguments assumed to be not in $\max_i$. That means we get the same normal form if the cyclic successor functions succ_i, $i \in [d]$, are replaced by successor functions for which $\mathrm{succ}_i(a) = a$ (instead of $a^{(i)}$) if $a \in \max_i$.*

Using the normal form so obtained for $\mathrm{ESO}(\forall^1, \text{arity } 1)$ formulas on pixel structures, it is rather easy to prove the following equivalence.

Proposition 3.9. *For any $d > 0$ and any d-language L on Σ: $L \in \mathrm{REC}^d \Leftrightarrow \mathsf{pixel}^d(L) \in \mathrm{ESO}(\forall^1, \text{arity } 1)$.*

Proof. $\boxed{\Rightarrow}$ A picture belongs to L if there exists a tiling of its domain whose projection coincides with the content of it cells. In the logic involved in the proposition, the "arity 1" corresponds to formulating the existence of the tiling, while the "$\forall^1$" is the syntactic resource needed to express that the tiling behaves as expected. Let us detail these considerations.

By Definition 3.2, there exist an alphabet Γ (which can be assumed disjoint from Σ), a surjective function $\pi : \Gamma \to \Sigma$ and d subsets $\Delta_1, \dots, \Delta_d \subseteq (\Gamma^\sharp)^2$ such that

$$L = \{\pi \circ p' : p' \in L(\Delta_1, \dots, \Delta_d)\} \tag{4}$$

The membership of a picture $p' : [n]^d \to \Gamma$ to $L(\Delta_1, \dots, \Delta_d)$ is easily expressed on $\mathsf{pixel}^d(p')$ by a first-order formula that asserts, for each dimension $i \in [d]$, that, for any pixel x of p', the couple $(x, \mathrm{succ}_i(x))$ can be tiled with some element of Δ_i. Namely,

$p' \in L(\Delta_1, \dots, \Delta_d)$ iff $\mathsf{pixel}^d(p') \models \forall x \psi_{\Delta_1,\dots,\Delta_d}(x)$, where:

$$\psi_{\Delta_1,\dots,\Delta_d}(x) = \bigwedge_{i\in[d]} \left\{ \begin{array}{lll} \min_i(x) \to & \bigvee_{(\sharp,s)\in\Delta_i} Q_s(x) & \wedge \\ \neg\max_i(x) \to & \bigvee_{(s,s')\in\Delta_i} (Q_s(x) \wedge Q_{s'}(\mathrm{succ}_i(x))) & \wedge \\ \max_i(x) \to & \bigvee_{(s,\sharp)\in\Delta_i} Q_s(x) & \end{array} \right\}.$$

Now, by (4), a picture $p : [n]^d \to \Sigma$ belongs to L iff it results from a π-renaming of a picture $p' \in L(\Delta_1, \dots, \Delta_d)$. It means there exists a Γ-labeling of p (that is, a tuple $(Q_s)_{s\in\Gamma}$ of subsets of $[n]^d$) corresponding to a picture of $L(\Delta_1, \dots, \Delta_d)$ (i.e. fulfilling $\forall x \psi_{\Delta_1,\dots,\Delta_d}(x, (Q_s)_{s\in\Gamma})$) and from which the actual Σ-labeling of p (that is, the subsets $(Q_s)_{s\in\Sigma}$) is obtained *via* π. More precisely:

$p \in L$ iff $\mathsf{pixel}^d(p) \models \Theta_L$, where:

$$\Theta_L = \ (\exists Q_s)_{s\in\Gamma}\ \forall x : \psi_{\Delta_1,\dots,\Delta_d}(x) \ \wedge \bigwedge_{s\in\Sigma} \left(Q_s(x) \to [\bigoplus_{s'\in\pi^{-1}(s)} Q_{s'}(x) \wedge \bigwedge_{s'\in\Gamma\setminus\pi^{-1}(s)} \neg Q_{s'}(x)\ ] \right).$$

Here, $\bigoplus$ denotes the exclusive disjunction. Notice that since $\Sigma \cap \Gamma = \emptyset$, the tuples $(Q_s)_{s\in\Sigma}$ and $(Q_s)_{s\in\Gamma}$ are also disjoint. As Θ_L clearly belongs to $\mathrm{ESO}(\forall^1, \text{arity } 1)$, the proof is complete.

$\boxed{\Leftarrow}$ Consider L such that $\mathsf{pixel}^d(L) \in \mathrm{ESO}(\forall^1, \text{arity } 1)$. Lemma 3.7 ensures that $\mathsf{pixel}^d(L)$ is characterized by a sentence of the form:

$$\exists \mathbf{U} \forall x \bigwedge_{i\in[d]} \left\{ \begin{array}{rcll} \min_i(x) & \to & \mathsf{m}_i(x) & \wedge \\ \max_i(x) & \to & \mathsf{M}_i(x) & \wedge \\ \neg\max_i(x) & \to & \Psi_i(x) & \end{array} \right\}. \tag{5}$$

Here, $\mathbf{U}$ is a list of monadic relation variables and m_i, M_i, Ψ_i are quantifier-free formulas such that atoms of m_i and M_i have all the form $Q(x)$ and atoms of Ψ_i have all the form $Q(x)$ or $Q(\mathrm{succ}_i(x))$, wih $Q \in \{(Q_s)_{s\in\Sigma}, \mathbf{U}\}$.

We have to prove that L is the projection of some local d-language L_{loc} on some alphabet Γ, that is a $(\Delta_1,\dots, \Delta_d)$-tiled language for some $\Delta_1,\dots, \Delta_d \subseteq \Gamma^2$. Let $U_1, \dots, U_k$ denote the list of (distinct) elements of the set $\{(Q_s)_{s\in\Sigma}, \mathbf{U}\}$ of unary relation symbols of φ, so that the first ones $U_1, \dots, U_m$ are the Q_s's (here, $\min_i$ and $\max_i$ symbols are excluded). The trick is to put each subformula $\mathsf{m}_i(x)$, $\mathsf{M}_i(x)$ and $\Psi_i(x)$ of φ into its *complete disjunctive normal form* with respect to $U_1,\dots,U_k$. Typically, each subformula $\Psi_i(x)$ whose atoms are of the form $U_j(x)$ or $U_j(\mathrm{succ}_i(x))$, for some $j \in [k]$, is transformed into the following "complete disjunctive normal form":

$$\bigvee_{(\epsilon,\epsilon')\in\Delta_i} \left(\bigwedge_{j\in[k]} \epsilon_j U_j(x) \wedge \bigwedge_{j\in[k]} \epsilon'_j U_j(\mathrm{succ}_i(x)) \right). \tag{6}$$

Here, the following conventions are adopted:

- $\epsilon = (\epsilon_1,\dots,\epsilon_k) \in \{0,1\}^k$ and similarly for ϵ';
- for any atom α and any bit $\epsilon_j \in \{0,1\}$, $\epsilon_j\alpha$ denotes the literal α if $\epsilon_j = 1$, and the literal $\neg\alpha$ otherwise.

For $\epsilon \in \{0,1\}^k$, we denote by $\Theta_\epsilon(x)$ the "complete conjunction" $\bigwedge_{j\in[k]} \epsilon_j U_j(x)$. Intuitively, $\Theta_\epsilon(x)$ is a complete description of x and the set

$$\Gamma = \bigcup_{i\in[m]} \{0^{i-1}10^{m-i}\} \times \{0,1\}^{k-m}$$

is the set of possible colors (remember that the Q_s's that are the U_j's for $j \in [m]$ form a partition of the domain). The complete disjunctive normal form (6) of $\Psi_i(x)$ can be written into the suggestive form

$$\bigvee_{(\epsilon,\epsilon')\in\Delta_i} (\Theta_\epsilon(x) \wedge \Theta_{\epsilon'}(\mathrm{succ}_i(x))).$$

If each subformula $\mathsf{m}_i(x)$ and $\mathsf{M}_i(x)$ of φ is similarly put into complete disjunctive normal form, that is $\bigvee_{(\sharp,\epsilon)\in\Delta_i} \Theta_\epsilon(x)$ and $\bigvee_{(\epsilon,\sharp)\in\Delta_i} \Theta_\epsilon(x)$, respectively (there is no ambiguity in our implicit definition of the sets Δ_i, since $\sharp \notin \Gamma$), then the above sentence (5) equivalent to φ becomes the following equivalent sentence:

$$\varphi' = \exists \mathbf{U} \forall x \bigwedge_{i\in[d]} \left\{ \begin{array}{lcll} \min_i(x) & \to & \bigvee_{(\sharp,\epsilon)\in\Delta_i} \Theta_\epsilon(x) & \wedge \\ \max_i(x) & \to & \bigvee_{(\epsilon,\sharp)\in\Delta_i} \Theta_\epsilon(x) & \wedge \\ \neg\max_i(x) & \to & \bigvee_{(\epsilon,\epsilon')\in\Delta_i} (\Theta_\epsilon(x) \wedge \Theta_{\epsilon'}(\mathrm{succ}_i(x))) & \end{array} \right\}$$

Finally, let L_{loc} denote the d-language over Γ defined by the first-order sentence φ_{loc} obtained by replacing each Θ_ϵ by the new unary relation symbol Q_ϵ in the first-order part of φ'. In other words, $\mathsf{pixel}^d(L_{\mathrm{loc}})$ is defined by the following first-order sentence:

$$\varphi_{\mathrm{loc}} = \forall x \bigwedge_{i\in[d]} \left\{ \begin{array}{lcll} \min_i(x) & \to & \bigvee_{(\sharp,\epsilon)\in\Delta_i} Q_\epsilon(x) & \wedge \\ \max_i(x) & \to & \bigvee_{(\epsilon,\sharp)\in\Delta_i} Q_\epsilon(x) & \wedge \\ \neg\max_i(x) & \to & \bigvee_{(\epsilon,\epsilon')\in\Delta_i} (Q_\epsilon(x) \wedge Q_{\epsilon'}(\mathrm{succ}_i(x))) & \end{array} \right\}$$

Hence, $L_{\mathrm{loc}} = L(\Delta_1,\dots,\Delta_d)$. That is, L_{loc} is indeed local and the corresponding sets of tiles are the Δ_i's of the previous formula. It is now easy to see that our initial d-language L is the projection of the local language L_{loc} by the projection $\pi : \Gamma \to \Sigma$ defined as follows: $\pi(\epsilon) = s$ iff $\epsilon_i = 1$ for $i \in [m]$ and U_i is Q_s. This completes the proof. $\square$

3.3. A normalization of ESO(arity 1) on pixel structures

Let us now come to the last step of the proof of Theorem 3.6. A key point of this step is a quantifier elimination result for first-order logic, proved independently in [8] and in [46]. Its statement needs two new definitions.

Definition 3.10. *A bijective structure is a finite structure of the form*

$$\mathcal{S} = (dom(\mathcal{S}), f_1, \ldots, f_d, f_1^{-1}, \ldots, f_d^{-1}, U_1, \ldots, U_m),$$

where each f_i is a unary bijective function of inverse bijection f_i^{-1} and the U_i's are unary relations.

Definition 3.11. *A cardinality formula is a first-order formula of the form $\exists^{\geq k} x \psi(x)$, where $k \geq 1$ and $\psi(x)$ is a quantifier-free formula with only one variable x. The quantifier $\exists^{\geq k} x$ means "there exist at least k elements x such that".*

Let us present without proof the following normalization of first-order logic on bijective structures which was proved in [8, 46] and extends a pioneering result by Seese [63, 64].

Proposition 3.12 ([8, 46]). *On bijective structures, each first-order sentence is equivalent to a boolean combination of cardinality formulas.*

Clearly, a pixel structure expanded by the inverse functions of its successor functions, that is by the *predecessor* functions $\mathrm{pred}_i = \mathrm{succ}_i^{-1}$, $i \in [d]$, is a bijective structure. Therefore:

Corollary 3.13. *On pixel structures, each first-order sentence is equivalent to a boolean combination of cardinality formulas.*

Proof. By Proposition 3.12, each first-order sentence on pixel structures is equivalent to a boolean combination of cardinality formulas on pixel structures expanded by the predecessor functions. It is easy to see that any occurrence of a predecessor symbol pred_i in a formula $\exists^{\geq k} x\, \psi(x)$ can be eliminated: this can be done by replacing each occurrence of x in $\psi(x)$ by $\mathrm{succ}_i(x)$ (this is justified by the bijectivity of the successor function) and simplifying $\mathrm{pred}_i(\mathrm{succ}_i(x)) = x$ (this is justified by the fact that all the predecessor and successor functions commute each other). □

This allows to prove the following proposition.

Proposition 3.14. ESO(arity 1) $\subseteq$ ESO($\forall^1$, arity 1) *on d-pixel structures, for any dimension $d > 0$.*

Proof. Let $\exists \mathbf{U} \varphi$ be an ESO(arity 1)-sentence. It is sufficient to prove that its first-order part φ can be transformed into an equivalent formula in ESO($\forall^1$, arity 1). By Corollary 3.13, φ is equivalent to a boolean combination of sentences of the form $\psi^{\geq k} = \exists^{\geq k} x\, \psi(x)$ (for $k \geq 1$) where $\psi(x)$ is a quantifier-free formula with the single variable x. Therefore, it is easily seen that proving the proposition amounts to show that each sentence of the form $\psi^{\geq k}$ or $\neg\psi^{\geq k}$ can be translated in ESO($\forall^1$, arity 1) on pixel structures. This is done as follows.

For a given sentence $\exists^{\geq k} x\, \psi(x)$, we introduce k new unary relations $U^{=0}, U^{=1}, \ldots, U^{=k-1}$ and $U^{\geq k}$, with the intended meaning:

> A pixel $a \in [n]^d$ belongs to $U^{=j}$ (resp. $U^{\geq k}$) iff there are exactly j (resp. at least k) pixels $b \in [n]^d$ lexicographically smaller than or equal to a such that $\mathsf{pixel}^d(p) \models \psi(b)$.

Then, we have to compel these relation symbols to fit their expected interpretations by means of a universal first-order formula with a single variable. First, we demand the relations to form a partition of the domain:

(1) $\bigwedge_{i<j<k} \left(\neg U^{=i}(x) \vee \neg U^{=j}(x)\right) \wedge \bigwedge_{i<k} \left(\neg U^{=i}(x) \vee \neg U^{\geq k}(x)\right).$

Then, we temporarily denote by $\leq_{\text{lex}}$ the lexicographic order on $[n]^d$ inherited from the natural order on $[n]$, and by succ_{lex}, $\min_{\text{lex}}$, $\max_{\text{lex}}$ its associated successor function and unary relations corresponding to extremal elements. Then the sets described above can be defined inductively by the conjunction of the following six formulas:

(2) $(\min_{\text{lex}}(x) \wedge \neg\psi(x)) \to U^{=0}(x)$

(3) $(\min_{\text{lex}}(x) \wedge \psi(x)) \to U^{=1}(x)$

(4) $\bigwedge_{i<k} \left((\neg\max_{\text{lex}}(x) \wedge U^{=i}(x) \wedge \neg\psi(\text{succ}_{\text{lex}}(x))\right) \to U^{=i}(\text{succ}_{\text{lex}}(x))$

(5) $\bigwedge_{i<k-1} \left((\neg\max_{\text{lex}}(x) \wedge U^{=i}(x) \wedge \psi(\text{succ}_{\text{lex}}(x))\right) \to U^{=i+1}(\text{succ}_{\text{lex}}(x))$

(6) $\left((\neg\max_{\text{lex}}(x) \wedge U^{=k-1}(x) \wedge \psi(\text{succ}_{\text{lex}}(x))\right) \to U^{\geq k}(\text{succ}_{\text{lex}}(x))$

(7) $\left((\neg\max_{\text{lex}}(x) \wedge U^{\geq k}(x)\right) \to U^{\geq k}(\text{succ}_{\text{lex}}(x))$

Hence, under the hypothesis $(1) \wedge \ldots \wedge (7)$, the sentences $\psi^{\geq k}$ and $\neg\psi^{\geq k}$ are equivalent, respectively, to

$$\forall x : \max_{\text{lex}}(x) \to U^{\geq k}(x) \text{ and } \forall x : \max_{\text{lex}}(x) \to \neg U^{\geq k}(x).$$

To complete the proof, it remains to get rid of symbols succ_{lex}, $\min_{\text{lex}}$ and $\max_{\text{lex}}$ that are not allowed in our language. It is done by referring to these symbols implicitly rather than explicitly. For instance, since $\text{succ}_{\text{lex}}(x) = \text{succ}_i \text{succ}_{i+1} \ldots \text{succ}_d(x)$ for the smallest $i \in [d]$ such that $\bigwedge_{j>i} \max_j(x)$, each formula φ involving $\text{succ}_{\text{lex}}(x)$ actually corresponds to the conjunction:

$$\bigwedge_{i\in[d]} \left((\neg\max_i(x) \wedge \bigwedge_{i<j\leq d} \max_j(x)) \to \varphi_i \right),$$

where φ_i is obtained from φ by the substitution $\text{succ}_{\text{lex}}(x) \rightsquigarrow \text{succ}_i \ldots \text{succ}_d(x)$. Similar arguments allow to get rid of $\min_{\text{lex}}$ and $\max_{\text{lex}}$. □

Remark 3.15. *In this proof of* ESO(arity 1) $\subseteq$ ESO($\forall^1$, arity 1) *on pixel structures* $\mathsf{pixel}^d(p)$*, two crucial features of such a structure are involved:*

- *its* bijective *nature, which allows to rewrite first-order formulas as (boolean combinations of) cardinality formulas with a single first-order variable;*

- *the* regularity *of its predefined arithmetics (the functions $succ_i$ defined for each dimension), that endows $\mathsf{pixel}^d(p)$ with a grid structure: it enables us to implicitly define a linear order of the whole domain $dom(p)$ by means of first-order formulas with a single variable, which in turn allows to express cardinality formulas by "cumulative" arguments, via the sets $U^{=i}$ and $U^{\geq k}$.*

Proposition 3.14 straightforwardly generalizes to the various structures that fulfill these two properties.

To conclude this section, let us mention that we can rather easily derive from Theorem 3.6 the following additional characterization of REC^d:

Corollary 3.16. *For any $d > 0$ and any d-language L,*

$$L \in \mathrm{REC}^d \Leftrightarrow \mathsf{pixel}^d(L) \in \mathrm{ESO}(\mathrm{var}\,1).$$

4. Linear time of cellular automata and its logical characterization

Beside the notion of *recognizable* picture language, the main concept studied in this paper is the classical notion of *linear time* complexity on *nondeterministic cellular automata* of any dimension (see for instance [37, 67, 38, 7, 66, 54, 47, 55]). We first present some general and informal considerations about cellular automata. In Subsection 4.2, we will give a precise definition of a specific model, the *d-dimensional one-way* cellular automaton. Then, in Subsection 4.3, we will state our main logical characterization theorem and will prove its first implication.

4.1. General considerations about cellular automata

Cellular automata are the simplest model of local and massively parallel computation. Basically, a cellular automaton is a regular array of identical cells called *cellular array*. Each cell is a copy of the same finite state automaton. At each step of a computation, the next state of each cell is produced by a local transition rule according to the current states of its neighbor cells. There are various definitions of cellular automata, depending essentially on the form of the cellular array, in particular on its dimension, on the chosen *neighborhood*, and on how the input is given to the cellular array.

In general, the cellular array is a finite line of cells (dimension $d = 1$), or is a finite rectangular grid of some fixed dimension $d = 2, 3$, or more. So, at each step of a computation, the states of the cellular array constitute a rectangular picture of dimension d over the state alphabet. It is natural to assume that the input of a d-dimensional cellular automaton is a d-picture on an alphabet Σ included in the state alphabet.

Remark 4.1. *By definition, the space used by a d-dimensional cellular automaton is exactly the space (set of cells) occupied by its input d-picture.*

Recall that all input d-pictures are assumed of the form $p : [n]^d \to \Sigma$. The integer n is called the *side* of the picture p and is the reference parameter for measuring complexity.

In the literature (e.g. [37, 67, 38, 7, 55, 47]), one essentially uses two different kinds of neighborhoods (each with two variants):

- a *two-way* neighborhood: the neighbors of a cell $\mathbf{a} = (a_1, \ldots, a_d)$ are the cells $\mathbf{b} = (b_1, \ldots, b_d)$ such that $\max_{i\in[d]} |a_i - b_i| \leq 1$, this is the so-called Moore neighborhood (resp. $\Sigma_{i\in[d]} |a_i - b_i| \leq 1$, this is the von Neumann neighborhood);

- a *one-way* neighborhood [3] : the neighbors of a cell $\mathbf{a} = (a_1, \ldots, a_d)$ are the cells $\mathbf{b} = (b_1, \ldots, b_d)$ such that $b_i \geq a_i$, for each $i \in [d]$, and $\max_{i \in [d]} (b_i - a_i) \leq 1$(resp. $\Sigma_{i \in [d]} (b_i - a_i) \leq 1$).

A cellular automaton is called *two-way* or *one-way* according to its specified neighborhood. A d-dimensional cellular automaton runs within time $T(n)$ if any of its computations stops within exactly $T(n) - 1$ steps, for any input d-picture of side n. The most studied time bounded classes of picture languages are the following:

- the class of picture languages recognized by cellular automata in *linear time*, that means time $O(n)$;
- the class of picture languages recognized by cellular automata in *real time*, that means the minimal time necessary for that the content of every cell of the input picture can be communicated to the reference cell $1^d = (1, \ldots, 1)$: e.g., it is time $n + 1$ if the chosen neighborhood is one-way with the above-mentioned condition $\max_{i \in [d]} (b_i - a_i) \leq 1$.

As the other computational models, cellular automata can be *deterministic* or *nondeterministic*. In this paper, we are interested in the *nondeterministic* case. While most relationships between the above complexity classes, linear/real time on one-way/two-way cellular automata, are open questions in the deterministic case (e.g., see [55] and the nice survey [67] that describes what is known in the deterministic case), the situation is much simpler and well-known in the nondeterministic case, as expressed by the following proposition given here without proof.

Proposition 4.2 (folklore). *For any dimension d and every d-picture language L, the following conditions are equivalent:*

1. *L is recognized in* linear time *by a nondeterministic d-dimensional* two-way *cellular automaton;*
2. *L is recognized in* linear time *by a nondeterministic d-dimensional* one-way *cellular automaton;*
3. *L is recognized in* real time *by a nondeterministic d-dimensional* two-way *cellular automaton (with either above Moore or von Neumann neighborhoods);*
4. *L is recognized in* real time *by a nondeterministic d-dimensional* one-way *cellular automaton (with either above neighborhoods).*

So, the linear time complexity class of *nondeterministic* cellular automata is a very *robust* notion; in particular, as it is equal to the real time class, it is the minimal time complexity class that allows to synchronize, i.e. to communicate to a reference cell the content of each other cell.

4.2. Linear time of nondeterministic cellular automata of any dimension

For simplicity of notation, we only present formally the notion of *one-way* d-dimensional cellular automaton, instead of the more usual notion of *two-way* d-dimensional cellular automaton. There are some technicalities in our definition of the transition function of such an automaton. This is due to the need to distinguish the different possible positions of the pixels of a picture w.r.t. its border: the chosen one-way neighborhood of a cell $\mathbf{x} = (x_1, \ldots, x_d)$, that is the set of cells $\mathbf{y} = (y_1, \ldots, y_d)$ such that $0 \leq y_i - x_i \leq 1$ for each $i \in [d]$, may be incomplete according to the *position* of the cell $\mathbf{x}$ w.r.t. the border of the picture. This is defined as follows.

[3] It is called *one-way* because each information can only be communicated along each coordinate in *one* direction, here the decreasing direction. At the opposite, each information can be communicated in *both* decreasing *and* increasing directions in the *two-way* case.

Definition 4.3. *(See Figure 1.) A pixel* $\mathbf{x} = (x_1, \ldots, x_d) \in [n]^d$ *is* in position $\mathbf{a} = (a_1, \ldots, a_d) \in \{0,1\}^d$ *in a picture* $p : [n]^d \to \Gamma$ *or in the domain* $[n]^d$ *if for all* $i \in [d]$ *we have* $a_i = 0$ *if* $x_i = n$ *and* $a_i = 1$ *if* $x_i < n$.

We are going to define the transition function on a pixel $\mathbf{x}$ of a picture p according to some "neighborhood" denoted $p_{\mathbf{a},\mathbf{x}}$ (it is a subpicture of p) whose domain, denoted by $\mathsf{Dom}_{\mathbf{a}}$, depends on the position $\mathbf{a}$ of the pixel in the picture.

Definition 4.4. *For each* $\mathbf{a} = (a_1, \ldots, a_d) \in \{0,1\}^d$*, let us define the* $\mathbf{a}$-domain *as* $\mathsf{Dom}_{\mathbf{a}} = [0, a_1] \times \cdots \times [0, a_d]$.

The $\mathbf{a}$-neighborhood *of some pixel* $\mathbf{x} \in [n]^d$ *in position* $\mathbf{a}$ *in a picture* $p : [n]^d \to \Gamma$ *is the function* $p_{\mathbf{a},\mathbf{x}} : \mathsf{Dom}_{\mathbf{a}} \to \Gamma$ *defined as* $p_{\mathbf{a},\mathbf{x}}(b) = p(\mathbf{x} + \mathbf{b})$*, where* $\mathbf{x} + \mathbf{b}$ *denotes the sum of the vectors* $\mathbf{x}$ *and* $\mathbf{b}$.

We denote by $\mathsf{neighb}_a(\Gamma)$ *the set of all possible* $\mathbf{a}$*-neighborhoods on an alphabet* Γ*, that is the set of functions* $v : \mathsf{Dom}_{\mathbf{a}} \to \Gamma$.

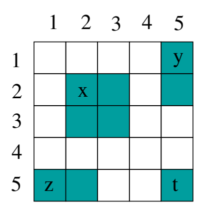

Figure 1: pixels $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ and $\mathbf{t}$ are, respectively, in position $(1,1)$, $(0,1)$, $(1,0)$, and $(0,0)$. Whence their associated neighborhoods, which appear as colored pixels on the figure.

Such neighborhoods are used to describe the "transition function" of the cellular automata that we now define:

Definition 4.5. *A* one-way nondeterministic d-dimensional cellular automaton *(*d-automaton*, for short) over an alphabet* Σ *is a tuple* $\mathcal{A} = (\Sigma, \Gamma, \delta, F)$*, where*

- *the finite alphabet* Γ *called the* set of states *of* $\mathcal{A}$ *includes the* input alphabet Σ *and the set* F *of* accepting states*:* $\Sigma, F \subseteq \Gamma$*;*
- δ *is the (nondeterministic)* transition function *of* $\mathcal{A}$*: it is a family of* $\mathbf{a}$-transition functions $\delta = (\delta_{\mathbf{a}})_{\mathbf{a} \in \{0,1\}^d}$ *of the form* $\delta_{\mathbf{a}} : \mathsf{neighb}_{\mathbf{a}}(\Gamma) \to \mathcal{P}(\Gamma)$.

Let us now define a computation.

Definition 4.6. *Let* $\mathcal{A} = (\Sigma, \Gamma, \delta, F)$ *be a d-automaton and* $p, p' : [n]^d \to \Gamma$ *be two d-pictures on* Γ*. We say that* p' *is a* successor *of* p *for* $\mathcal{A}$*, denoted by* $p' \in \mathcal{A}(p)$*, if, for each position* $\mathbf{a} \in \{0,1\}^d$ *and each point* $\mathbf{x}$ *of position* $\mathbf{a}$ *in* $[n]^d$*,* $p'(\mathbf{x}) \in \delta_{\mathbf{a}}(p_{\mathbf{a},\mathbf{x}})$*. The set of* j^{th}-successors *of* p *for* $\mathcal{A}$*, denoted by* $\mathcal{A}^j(p)$*, is defined inductively:*

$$\mathcal{A}^0(p) = \{p\} \text{ and, for } j \geq 0,\ \mathcal{A}^{j+1}(p) = \bigcup_{p' \in \mathcal{A}^j(p)} \mathcal{A}(p').$$

Definition 4.7. *A* computation *of a d-automaton* $\mathcal{A} = (\Sigma, \Gamma, \delta, F)$ *on an input d-picture* $p : [n]^d \to \Sigma$ *is a sequence* $p_1, p_2, p_3, \ldots$ *of d-pictures such that* $p_1 = p$ *and* $p_{i+1} \in \mathcal{A}(p_i)$ *for each i. The picture* p_i, $i \geq 1$, *is called the* i^{th} configuration *of the computation. A computation is* accepting *if it is finite – it has the form* $p_1, p_2, \ldots, p_k$ *for some k – and the cell of minimal coordinates,* $1^d = (1, \ldots, 1)$, *of its last configuration is in an accepting state:* $p_k(1^d) \in F$.

Definition 4.8. *Let* $\mathcal{A} = (\Sigma, \Gamma, \delta, F)$ *be a d-automaton and let* $T : \mathbb{N} \to \mathbb{N}$ *be a function such that* $T(n) > n$. *A d-picture p on* Σ *is* accepted by $\mathcal{A}$ in time $T(n)$ *if* $\mathcal{A}$ *admits an accepting computation of length* $T(n)$ *on p. That means there exists a computation* $p = p_1, p_2, \ldots, p_{T(n)} \in \mathcal{A}^{T(n)-1}(p)$ *of* $\mathcal{A}$ *on p such that* $p_{T(n)}(1^d) \in F$.

A d-language L on Σ *is* accepted, *or* recognized, *by* $\mathcal{A}$ in time $T(n)$ *if it is the set of d-pictures p accepted by* $\mathcal{A}$ in time $T(n)$, *i.e. such that there exists* $p' \in \mathcal{A}^{T(n)-1}(p)$ *where n is the size of p and with* $p'(1^d) \in F$.

If $T(n) = cn + c'$, *for some integers* c, c', *then L is said to be* recognized in linear time *and we write* $L \in \mathrm{NLIN}^d_{\mathrm{ca}}$.

The time bound $T(n) > n$ of the above definition is necessary and sufficient to allow the information of any pixel of p to be communicated to the pixel of minimal coordinates, 1^d.

Remark 4.9. *As stated in Proposition 4.2 above, the nondeterministic linear time class* NLIN^d_{ca} *is very robust, i.e. is not modified by many changes in the definition of the automaton or in its time bound. In particular, the constants* c, c' *defining the bound* $T(n) = cn + c'$ *can be arbitrarily fixed, provided* $T(n) > n$. *For example, the class* NLIN^d_{ca} *does not change if we take the* minimal *time* $T(n) = n + 1$, *called* real time.

4.3. A logical characterization of nondeterministic linear time for cellular automata

With Theorem 3.6, we have stated a logical characterization of REC, the class of recognizable picture languages. The four forthcoming sections (including the present one) are devoted to the second central issue of this paper: a logical characterization of $\mathrm{NLIN}_{\mathrm{ca}}$, the linear time complexity class of nondeterministic cellular automata. To be precise, we will soon establish:

Theorem 9.1. *For any* $d > 0$ *and any d-language L,*

$$L \in \mathrm{NLIN}^d_{ca} \Leftrightarrow \mathsf{coord}^d(L) \in \mathrm{ESO}(\mathrm{var}\, d+1).$$

By abuse of language, we will write "$\mathrm{NLIN}^d_{ca} = \mathrm{ESO}(\mathrm{var}\, d+1)$ *on d-coordinate structures" to mention the above equivalence.*

There are many such logical characterizations of complexity classes, see *e.g.* [11, 25, 42], or the surveys [34, 14]. In most cases, the right-to-left implication, i.e. the inclusion of the logically defined class in the complexity class, is easier than the converse one. This is due to the great flexibility of computational models, that allows to easily translate, in most cases, syntactic properties of the logic into bounds over computational resources. As a significant example, in the proof of Fagin's statement that ESO $\subseteq$ NP [11]:

- existential second-order quantifications relate to nondeterminism;
- universal first-order quantifications give rise to deterministic for loops;
- arities of relation symbols are interpreted in terms of the size of handled objects;
- in particular, arities of existentially quantified relation symbols determine the size of objects to be guessed, that is, somehow, the "amount" of nondeterminism involved in the computation.

Those correspondences hold even more tightly when computation resources are tighter : see for instance the proof of the inclusion SO(HORN) $\subseteq$ P by Grädel [25], or that of ESOF(var 1) $\subseteq$ NLIN by the present authors [27, 28]. However, in either of those two cases some preliminary normalization result about the logic, actually SO(HORN) $\subseteq$ ESO(HORN) and ESOF(var 1) $\subseteq$ ESOF($\forall^1$, arity 1), respectively, appears as a crucial point of the proof; in either case, it is the condition to make logical formulas "handleable" for the computation model that has to evaluate them.

At the opposite of most cases, the right-to-left implication of Theorem 9.1, i.e. the inclusion ESO(var $d+1$) $\subseteq \mathrm{NLIN}^d_{\mathrm{ca}}$, is far more difficult to demonstrate than the converse one: it is a very hard task to prove that an ESO(var $d+1$)-sentence can be evaluated, over a d-coordinate structure taken as input, by a linear time cellular automaton. This is mainly due to the "local" behavior of cellular automata, which seems not adapted to the evaluation of an ESO(var $d+1$)-formula over a d-picture. Indeed, such a formula possibly connects pixels of the picture that may be arbitrarily far away from each other; and dealing with pixels that do not belong to a same neighborhood is seemingly out of the ability of our computational device. As outlined above, the proof of the right-to-left implication of Theorem 9.1 compels us to normalize the logic under consideration, in such a way that the formulas to be evaluated are rewritten under a form that can be "handled" by a cellular automaton. This will be done in several steps. We will cope with these successive normalizations in the forthcoming sections. For now, let us establish the "easy part" of the theorem, with Proposition 4.10 below.

Proposition 4.10. *For any $d > 0$ and any d-language L,*

$$L \in \mathrm{NLIN}^d_{ca} \Rightarrow \mathsf{coord}^d(L) \in \mathrm{ESO}(\mathrm{var}\, d+1).$$

Proof. Let $L \in \mathrm{NLIN}^d_{\mathrm{ca}}$. By Proposition 4.2, we can assume without loss of generality that L is recognized by a d-automaton $\mathcal{A} = (\Sigma, \Gamma, \delta, F)$ in real time, i.e. in time $n+1$. The ESO(var $d+1$)-sentence to be constructed is of the form $\exists (R_s)_{s\in\Gamma} \forall \mathbf{x} \forall t\, \psi(\mathbf{x}, t)$, where $\psi(\mathbf{x}, t)$ is a quantifier-free formula such that:

- ψ uses a list of exactly $d+1$ first-order variables $\mathbf{x} = (x_1, \dots, x_d)$ and t. Intuitively, the d first ones represent the coordinates of any point in $dom(p) = [n]^d$ and the last one represents any of the first n instants $t \in [n]$ of the computation (the last instant $n+1$ is not explicitly represented);

- ψ uses, for each state $s \in \Gamma$, a relation symbol R_s of arity $d+1$. Intuitively, $R_s(a_1, \dots, a_d, t)$ holds, for any $\mathbf{a} = (a_1, \dots, a_d) \in [n]^d$ and any $t \in [n]$, iff the state of cell $\mathbf{a}$ at instant t is s.

- $\psi(\mathbf{x}, t)$ is the conjunction $\psi(\mathbf{x}, t) = \textsc{init}(\mathbf{x}, t) \wedge \textsc{step}(\mathbf{x}, t) \wedge \textsc{end}(\mathbf{x}, t)$ of three formulas whose intuitive meaning is the following.

- $\forall \mathbf{x} \forall t\ \textsc{init}(\mathbf{x}, t)$ describes the first configuration of $\mathcal{A}$, i.e. at initial instant 1, that is the input picture $p_1 = p$;
- $\forall \mathbf{x} \forall t\ \textsc{step}(\mathbf{x}, t)$ describes the computation between the instants t and $t+1$, for $t \in [n-1]$, i.e. describes the $(t+1)^{th}$ configuration p_{t+1} from the t^{th} one p_t, i.e. says $p_{t+1} \in \mathcal{A}(p_t)$;
- $\forall \mathbf{x} \forall t\ \textsc{end}(\mathbf{x}, t)$ expresses that the n^{th} configuration p_n leads to a (last) $(n+1)^{th}$ configuration $p_{n+1} \in \mathcal{A}(p_n)$ which is accepting, i.e. with an accepting state in cell 1^d: $p_{n+1}(1^d) \in F$.

Let us give explicitly these three formulas. The first one is straightforward:

$$\textsc{init}(\mathbf{x}, t) \equiv \min(t) \rightarrow \left\{ \bigwedge_{s\in\Sigma} (R_s(\mathbf{x}, t) \leftrightarrow Q_s(\mathbf{x}))\ \wedge \bigwedge_{s\in\Gamma\setminus\Sigma} \neg R_s(\mathbf{x}, t) \right\}$$

The second formula is

$$\textsc{step}(\mathbf{x},t) \equiv \bigwedge_{\mathbf{a}\in\{0,1\}^d} \bigwedge_{v\in\mathsf{neighb}_a(\Gamma)} \left\{ \left(\neg\max(t) \wedge P_{\mathbf{a}}(\mathbf{x}) \wedge \bigwedge_{b\in\mathsf{Dom}_{\mathbf{a}}} R_{v(b)}(\mathbf{x}+b,t) \right) \to \bigoplus_{s\in\delta_a(v)} R_s(\mathbf{x},\mathrm{succ}(t)) \right\}$$

Here, $\bigoplus$ denotes the exclusive disjunction. Furthermore:

- For $\mathbf{x} \in [n]^d$ and $\mathbf{a} = (a_1, \ldots, a_d) \in \{0,1\}^d$, the formula $P_{\mathbf{a}}(\mathbf{x})$ claims that the pixel $\mathbf{x}$ is in position $\mathbf{a}$. Namely: $P_{\mathbf{a}}(\mathbf{x}) \equiv \bigwedge_{i\in[d]} (\neg_i)\max(x_i)$, where $(\neg_i)$ is $\neg$ if $a_i = 1$, and *nothing* otherwise.
- For $\mathbf{b} = (b_1, \ldots, b_d) \in \{0,1\}^d$, $\mathbf{x}+\mathbf{b}$ abbreviates the tuple of terms $(\theta_1, \ldots, \theta_d)$ where, for each i, the term θ_i is x_i if $b_i = 0$, and $\mathrm{succ}(x_i)$ otherwise.

It is easy to verify that the formula $\forall\mathbf{x}\ \textsc{step}(\mathbf{x},t)$ means $p_{t+1} \in \mathcal{A}(p_t)$.

Finally, here is the last formula :

$$\textsc{end}(\mathbf{x},t) \equiv \left\{ \max(t) \wedge \bigwedge_{i\in[d]} \min(x_i) \right\} \to \left\{ \begin{array}{l} \left(\min(t) \to \bigvee_{v\in N_0} R_{v(0^d)}(\mathbf{x},t)\right) \wedge \\ \left(\neg\min(t) \to \bigvee_{v\in N_1} \bigwedge_{b\in\{0,1\}^d} R_{v(b)}(\mathbf{x}+b,t)\right) \end{array} \right\}$$

In this formula, the sets N_0 and N_1 are defined by:

$$N_0 = \{v \in \mathsf{neighb}_{0^d}(\Gamma) : \delta_{0^d}(v) \cap F \neq \emptyset\};\ N_1 = \{v \in \mathsf{neighb}_{1^d}(\Gamma) : \delta_{1^d}(v) \cap F \neq \emptyset\}.$$

Let us explain the meaning of the formula $\textsc{end}(\mathbf{x},t)$ which is rather technical due to the need to distinguish two cases according to the size n of the input picture: if $n = 1$ then the position of the reference pixel 1^d is 0^d; otherwise, i.e. if $n > 1$, it is 1^d. Under the hypothesis $\max(t)$, the condition $\min(t)$ (resp. $\neg\min(t)$) is equivalent to $n = 1$ (resp. $n > 1$). In either case, the disjunction $\bigvee_{v\in N_0}$ (resp. $\bigvee_{v\in N_1}$) expresses there exists $p_{n+1} \in \mathcal{A}(p_n)$ such that $p_{n+1}(1^d) \in F$.

Therefore, we have proved that, for any d-picture p on Σ, the structure $\mathsf{coord}^d(p)$ satisfies the $\mathrm{ESO}(\mathrm{var}\, d+1)$-sentence $\exists (R_s)_{s\in\Gamma} \forall\mathbf{x}\forall t\, \psi(\mathbf{x},t)$ if and only if $\mathcal{A}$ has an accepting computation $p = p_1, p_2, \ldots, p_{n+1}$ of length $n+1$ on p, i.e. $\mathcal{A}$ accepts p in time $n+1$, or, by definition, $p \in L$. Hence, $L \in \mathrm{ESO}(\mathrm{var}\, d+1)$, as required. ◻

Let us conclude this section with three remarks.

Remark 4.11. *The proof of Proposition 4.10 is almost straightforward* because*, by Proposition 4.2, it is* sufficient *that our logic can define an accepting computation of a* one-way *cellular d-dimensional cellular automaton of* real time $n+1$*, which is, up to* 1*, the side n of the input picture. In contrast, proving directly that an accepting computation of* linear time $O(n)$ *of a (more natural)* two-way *d-dimensional cellular automaton can be defined in* $\mathrm{ESO}(\mathrm{var}\, d+1)$ *would be possible but much more technical. Therefore, similarly to normalizations of logically defined classes, the robustness property expressed by Proposition 4.2 can be regarded as a "normalization" of the linear time complexity class of nondeterministic cellular automata.*

Remark 4.12. *One observes that the formula constructed in the proof of Proposition 4.10 belongs to the logic* $\mathrm{ESO}(\forall^{d+1}, \mathrm{arity}\, d+1)$ *that is seemingly more restricted than* $\mathrm{ESO}(\mathrm{var}\, d+1)$*. Actually, we will prove in the next section that these two logics coincide on coordinate d-structures.*

Remark 4.13. *The respective roles of time and space are seemingly dissymmetric in the sentence we have just constructed. However, the proof and the meaning of the converse implication* $\mathsf{coord}^d(L) \in \mathrm{ESO}(\mathrm{var}\, d+1) \Rightarrow L \in \mathrm{NLIN}^d_{ca}$ *that will be presented in the next sections show that actually the d dimensions of space are – or can be made – symmetrical w.r.t. time.*

5. A first normalization of ESO sentences with d variables on coordinate structures

From now on, we aim at establishing the converse of Proposition 4.10. It amounts to prove that for each $\mathrm{ESO}(\mathrm{var}\, d)$-formula, there exists a d-automaton able to evaluate this formula on any $(d-1)$-coordinate structure in linear time. As mentioned before Proposition 4.10, this result necessitates a preliminary normalization of the logic $\mathrm{ESO}(\mathrm{var}\, d)$. Indeed, the main feature of a cellular automaton is its ability to perform parallel computations on the pixels of an input picture, *as far as the data to be considered when dealing with a pixel are locally supplied*, that is, can be collected in a neighborhood of the pixel. But it appears that the properties encoded by an $\mathrm{ESO}(\mathrm{var}\, d)$-formula Φ do not fulfill this last prerequisite. For instance, assume that Φ is the $\mathrm{ESO}(\mathrm{var}\, 2)$-formula $\forall x, y : U(x,y) \leftrightarrow U(y,x)$. In order to evaluate Φ on a picture p, $\mathcal{A}$ has to check, for every couple (x,y), whether pixels of coordinates (x,y) and (y,x) have the same "color" with respect to U. That is, $\mathcal{A}$ has to check a property that relates pixels far from each other. And such a test seems to exceed the capacity of $\mathcal{A}$.

It should now be clear that making this evaluation possible means preventing that the cellular automaton has to check constraints connecting arbitrarily distant pixels. With this purpose, we must initially normalize the logic $\mathrm{ESO}(\mathrm{var}\, d+1)$, so as to force its formulas "to speak locally", that is, to assert local properties, which bring into play only adjacent cells, or at least pixels at a constant distance, namely, pixels belonging to the neighborhood of a given pixel $\mathbf{x}$.

We now go into this normalization task. The present section is dedicated to the proof of the following statement:

Theorem 5.1. *For any $d > 0$, $\mathrm{ESO}(\mathrm{var}\, d) = \mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$ on coordinate structures.*

This theorem will result from the forthcoming Proposition 5.2 and Proposition 5.4. The former states that each formula of $\mathrm{ESO}(\mathrm{var}\, d)$ can be written in such a way that its first-order part is prenex, universal, with no more than d universal quantifiers. With the latter, we rewrite each formula of $\mathrm{ESO}(\forall^d)$ in such a way that the arity of each guessed relation symbol of the formula is at most d.

5.1. Skolemization

Proposition 5.2. *For any $d > 0$, $\mathrm{ESO}(\mathrm{var}\, d) \subseteq \mathrm{ESO}(\forall^d)$ on coordinate structures of any dimension.*

Proof. The proof amounts to establishing that each $\mathrm{ESO}(\mathrm{var}\, d)$-formula is equivalent to an $\mathrm{ESO}(\forall^d)$-formula on any coordinate structure. Clearly, we can assume without loss of generality that the initial $\mathrm{ESO}(\mathrm{var}\, d)$-formula is first-order. So let's consider a first-order formula φ written with at most d variables. We first aim at writing φ under prenex form, *without introducing new first-order variables*. This entails introducing second-order variables existentially quantified. More precisely, the rewriting procedure is based on a depth-first traversal of the tree decomposition of φ. Each internal node of this tree corresponds to some subformula of φ of arity k – say $\theta(x_1, \ldots, x_k)$ –, and gives rise to a new relation symbol R_θ of the same arity. This relation is forced to encode the set $\{\mathbf{x} \text{ s.t. } \theta(\mathbf{x})\}$ *via* a formula $\mathrm{def}_\theta(R_\theta)$ defined as follows:

- If $\theta \equiv Qy\theta'(\mathbf{x}, y)$ where Q is a quantifier, then $\mathrm{def}_\theta(R_\theta) \equiv \forall \mathbf{x} : R_\theta(\mathbf{x}) \leftrightarrow Qy\theta'(\mathbf{x}, y)$
- If $\theta \equiv \theta'(\mathbf{x}) \circ \theta''(\mathbf{x})$ for some connective $\circ$, then $\mathrm{def}_\theta(R_\theta) \equiv \forall \mathbf{x} : R_\theta(\mathbf{x}) \leftrightarrow (\theta'(\mathbf{x}) \circ \theta''(\mathbf{x}))$.

If θ has no free variables, the relation symbol R_θ is choosen with arity 1 and its definition is written either $\forall x : R_\theta(x) \leftrightarrow Qy\theta(y)$ or $\forall x : R_\theta(x) \leftrightarrow (\theta' \circ \theta'')$, according to the form of θ. Here, x is any variable of φ distinct from y. Each time a node $\theta(\mathbf{x})$ has been visited, the corresponding R_θ and def_θ are generated and φ is updated by the substitution $\theta(\mathbf{x}) \rightsquigarrow R_\theta(\mathbf{x})$. Then, the procedure is run recursively on the formula so obtained.

Let us illustrate this procedure by running it on the first-order formula with three variables:

$$\varphi \equiv \exists x\,(\forall y \exists z\, U(x,y,z) \vee \exists y\, D(x,y)) \to \forall y\,(D(y,y) \vee \exists x\, U(x,y,x))\,. \tag{7}$$

We merely display the definition formulas generated by the procedure, along with the relation symbols $R_1,\dots,R_9$ corresponding to the nine internal nodes of φ. The successive updates of φ are implicit.

$$\begin{array}{llcl}
\mathrm{def}_1(R_1) \equiv \forall x,y : & R_1(x,y) & \leftrightarrow & \exists z\, U(x,y,z) \\
\mathrm{def}_2(R_2) \equiv \forall x : & R_2(x) & \leftrightarrow & \forall y\, R_1(x,y) \\
\mathrm{def}_3(R_3) \equiv \forall x : & R_3(x) & \leftrightarrow & \exists y\, D(x,y) \\
\mathrm{def}_4(R_4) \equiv \forall x : & R_4(x) & \leftrightarrow & (R_2(x) \vee R_3(x)) \\
\mathrm{def}_5(R_5) \equiv \forall y : & R_5(y) & \leftrightarrow & \exists x\, R_4(x) \\
\mathrm{def}_6(R_6) \equiv \forall y : & R_6(y) & \leftrightarrow & \exists x\, U(x,y,x) \\
\mathrm{def}_7(R_7) \equiv \forall y : & R_7(y) & \leftrightarrow & (D(y,y) \vee R_6(y)) \\
\mathrm{def}_8(R_8) \equiv \forall x : & R_8(x) & \leftrightarrow & \forall y\, R_7(y) \\
\mathrm{def}_9(R_9) \equiv \forall x : & R_9(x) & \leftrightarrow & (R_5(x) \to R_8(x))
\end{array}$$

Now, our initial formula can be rewritten:

$$\varphi \equiv \exists R_1,\dots,R_9 : \left(\bigwedge_{1\le i\le 9} \mathrm{def}_i(R_i) \right) \wedge \forall x\, R_9(x). \tag{8}$$

Notice that for each i, either def_i is prenex and universal, or it has the form $\forall \mathbf{u} : \alpha(\mathbf{u}) \leftrightarrow Qv\beta(\mathbf{u},v)$. It is easily seen that this last form is equivalent to the conjunction:

$$\forall \mathbf{u} Qv\,(\alpha(\mathbf{u}) \to \beta(\mathbf{u},v)) \;\wedge\; \forall \mathbf{u} Q^* v\,(\beta(\mathbf{u},v) \to \alpha(\mathbf{u})),$$

where Q^* is $\forall$ if Q is $\exists$ and *vice versa*. Therefore, following Equation (8), φ can now be written as a conjunction of prenex formulas, each of which involves no more than three variables and has a quantifier prefix of the shape $\forall \mathbf{x}$ or $\forall \mathbf{x} \exists y$. In order to write this conjunction under prenex form without adding new first-order variables, we have to "replace" existential quantifiers by universal ones. Afterward φ, as a conjunction of formulas of the type $\forall x,y,z\,\theta$, could be written under the requisite shape. We show below how to deal with this specific formula. The general case is strictly similar.

To get rid of existential quantifiers occuring in (some of) the def_i's, we will invoke the arithmetical symbols of the signature of φ. Remember that the conjuncts that are not still universal all have the form $\forall \mathbf{x} \exists y \theta(\mathbf{x},y)$, where $\mathbf{x}$ and y are tuples of first-order variables of respective arities k and 1. The predefined arithmetics included in coordinate signatures allow to defining, for any such conjonct, a relation of arity $k+1$ that witnesses the existence of some y fulfilling $\theta(\mathbf{x},y)$ for a given $\mathbf{x}$. This idea is completed as follows: Let W be a new $(k+1)$-ary relation symbol associated with $\exists y \theta(\mathbf{x},y)$. We want the assertion $\mathrm{W}(\mathbf{x},y)$ to signify that there exists $z \le y$ such that $\theta(\mathbf{x},z)$ holds. This interpretation is achieved thanks to the following formula:

$$\forall \mathbf{x} \forall y \left\{ \begin{array}{ll} & \min(y) \to (\mathrm{W}(\mathbf{x},y) \leftrightarrow \theta(\mathbf{x},y)) \\ \wedge & \mathrm{W}(\mathbf{x},\mathrm{succ}(y)) \leftrightarrow (\theta(\mathbf{x},\mathrm{succ}(y)) \vee \mathrm{W}(\mathbf{x},y)) \end{array} \right\}$$

We denote by $\mathrm{W} = \mathsf{witness}(\exists y \theta)$ this last formula. When it is satisfied, the assertion $\forall \mathbf{x} \exists y \theta(\mathbf{x},y)$ is clearly equivalent to $\forall \mathbf{x} \forall y : \max(y) \to \mathrm{W}(\mathbf{x},y)$.

For instance, the above formula $\mathrm{def}_1(R_1)$ gives rise to the non-universal formula $\mathrm{def}_1^1(R_1) \equiv \forall x,y \exists z : R_1(x,y) \to U(x,y,z)$. This shoud be managed as follows: A ternary relation symbol W_1 is introduced (i.e., existentially quantified) and compelled to fit its intended interpretation *via* the formula: $\mathrm{W}_1 = \mathsf{witness}(\delta_1)$, where $\delta_1 \equiv \exists z : R_1(x,y) \to U(x,y,z)$. Afterwards, the formula $\mathrm{def}_1^1(R_1)$ is replaced by $\forall x,y,z : \max(z) \to \mathrm{W}_1(x,y,z)$. When this task has been achieved for each non universal formula def_i, the formula displayed in (8) becomes:

$$\exists (R_i)_{i\in I}\ \exists (\mathrm{W}_j)_{j\in J} : \left(\bigwedge_{j\in J} \mathrm{W}_j = \mathsf{witness}(\delta_j)\right) \wedge \left(\bigwedge_{i\in I} \mathrm{def}_i\right) \wedge\ \forall x\, R_\ell(x).$$

Here, $I = \{1,\dots,9\}$, $J = \{1,3,5,6\}$ (the $j \in J$ correspond to formulas def_j that are non-universal), $\ell = 9$, and δ_j is the existential part of the (old) formula def_j. Clearly, this formula can be written in $\mathrm{ESO}(\forall^d)$ for $d = 3$. □

5.2. Arity vs number of first-order variables

We prove here a normalization of the logic $\mathrm{ESO}(\forall^d)$, similar to that of Proposition 2.8. This proof involves the following easy fact:

Fact 5.3. *Suppose we are given a family of functions $(f_i : X_i \to Y)_{i\in\mathcal{I}}$ and a family of relations $(R_i \subseteq X_i)_{i\in\mathcal{I}}$, indexed by the same finite set $\mathcal{I}$. The following assertions are equivalent:*

(i) $\forall i, j \in \mathcal{I}$, $\forall x \in X_i$, $\forall y \in X_j$: $f_i(x) = f_j(y) \Rightarrow R_i(x) = R_j(y)$;

(ii) $\exists R \subseteq Y$ *such that* $\forall i \in \mathcal{I}$, $\forall x \in Y$: $R_i(x) = R(f_i(x))$.

Proof. $(ii) \Rightarrow (i)$ is clear. For the converse implication, we define R on each set $f_i\langle X_i\rangle = \{f_i(x), x \in X_i\}$ by: $\forall x \in X_i$, $R(f_i(x)) = R_i(x)$. Hypothesis (i) guarantees the coherence of this definition. To complete it, we set $R(x) = 0$ for every $x \in Y \setminus \bigcup_{i\in\mathcal{I}} f_i\langle X_i\rangle$. The relation R thus defined clearly witnesses to condition (ii). □

Proposition 5.4. *For any $d > 1$, $\mathrm{ESO}(\forall^d) \subseteq \mathrm{ESO}(\forall^d, \text{arity } d)$ on coordinate structures.*

Proof. Let $\Phi \in \mathrm{ESO}(\forall^d)$. To fix ideas, let us assume that Φ has the very simple shape:

$$\Phi \equiv \exists R \forall x_1,\dots,x_d\, \varphi(\mathbf{x}, R), \tag{9}$$

where R is a *single* k-ary relation symbol for some $k > d$, and φ is a quantifier free formula. The formula to be built must have the form:

$$\Psi \equiv \exists \rho \forall x_1,\dots,x_d\, \psi(\mathbf{x}, \rho),$$

where ρ is a tuple of *d-ary* relation symbols and ψ is quantifier free.

The substitution of d-ary symbols for R rests in the limitation of the number of first-order variables in Φ: each atomic formula involving R has the form $R(t_1,\dots,t_k)$ where the t_i's are terms built on $x_1,\dots,x_d$. Therefore, although R is k-ary, *in each of its occurrences* it behaves as a d-ary symbol, dealing with the sole variables $x_1,\dots,x_d$. Hence, the key is to create a d-ary symbol for each occurrence of R in Φ or, more precisely, for each k-tuple of terms $(t_1,\dots,t_k)$ involved in a R-atomic formula.

More formally, let us denote by $\mathsf{T}(\Phi)$ the set of terms occuring in Φ, and by $\mathsf{T}_R(\Phi)$ the set of tuples of terms involved in a R-atomic subformula of Φ. That is, each element of $\mathsf{T}_R(\Phi)$ is a k-tuple

$$\mathbf{t}(\mathbf{x}) = (t_1(x_1,\dots,x_d),\dots,t_k(x_1,\dots,x_d)) \in \mathsf{T}(\Phi)^k$$

such that the formula $R(\mathbf{t}(\mathbf{x}))$ appears in Φ. For each $\mathbf{t}(\mathbf{x}) \in \mathsf{T}_R(\Phi)$, consider a new d-ary relation symbol $R_{\mathbf{t}(\mathbf{x})}$. Now, consider a σ-stucture S of domain $[n]$, and denote by $\langle S, R\rangle$ some expansion of S to $\sigma \cup \{R\}$. (That is, we denote by R both the relationnal symbol and its interpretation on $[n]$.) Furthermore, fix the S-interpretation of each $R_{\mathbf{t}(\mathbf{x})}$, for $\mathbf{t}(\mathbf{x}) \in \mathsf{T}_R(\Phi)$, by

$$\forall \mathbf{x} \in [n]^d : R_{\mathbf{t}(\mathbf{x})}(\mathbf{x}) = R(\mathbf{t}(\mathbf{x})), \tag{10}$$

and denote by $\mathbf{R}$ the tuple $(R_{\mathbf{t}(\mathbf{x})})_{\mathbf{t}(\mathbf{x}) \in \mathsf{T}_R(\Phi)}$ thus defined. Then clearly:

$$\langle S, R\rangle \models \forall \mathbf{x} \varphi(\mathbf{x}, R) \Leftrightarrow \langle S, \mathbf{R}\rangle \models \forall \mathbf{x} \tilde{\varphi}(\mathbf{x}, \mathbf{R}, \sigma), \tag{11}$$

where $\tilde{\varphi}$ is obtained from φ by substituing the formula $R_{\mathbf{t}(\mathbf{x})}(\mathbf{x})$ for each occurrence of the formula $R(\mathbf{t}(\mathbf{x}))$. Before continuing with this proof, let's illustrate the previous definitions with a simple example.

Example. Consider the ESO($\forall^2$)-formula: $\exists R \forall x, y \varphi(x, y, R)$, where

$$\varphi \equiv R(x, y, x) \wedge \neg R(y, x, y). \tag{12}$$

According to the notations used so far, we have: $d = 2$, $k = 3$, $\mathsf{T}(\Phi) = \{x, y\}$ and $\mathsf{T}_R(\Phi) = \{(x, y, x), (y, x, y)\}$. The binary relation symbols associated to the tuples of terms in $\mathsf{T}_R(\Phi)$ are denoted $R_{(x,y,x)}$ and $R_{(y,x,y)}$. The formula $\tilde{\varphi}$ obtained from φ, following (11), is written:

$$\tilde{\varphi} \equiv R_{(x,y,x)}(x, y) \wedge \neg R_{(y,x,y)}(x, y).$$

If, for any interpretation of R on $[n]$, we fix the interpretations of $R_{(x,y,x)}$ and $R_{(y,x,y)}$ as in (10):

$$\forall a, b < n : R_{(x,y,x)}(a, b) = R(a, b, a) \text{ and } R_{(y,x,y)}(a, b) = R(b, a, b),$$

then it is easily seen that $\langle S, R\rangle \models \forall x, y \varphi(x, y, R, \sigma)$ iff $\langle S, \mathbf{R}\rangle \models \forall x, y \tilde{\varphi}(x, y, R_{(x,y,x)}, R_{(y,x,y)}, \sigma)$. $\triangleleft$

Let's come back to the proof of Proposition 5.4. Equations (10) and (11) yield:

$$S \models \exists R \forall \mathbf{x} \varphi(\mathbf{x}, R) \Rightarrow S \models \exists \mathbf{R} \forall \mathbf{x} \tilde{\varphi}(\mathbf{x}, \mathbf{R}, \sigma), \tag{13}$$

where $\mathbf{R}$ is a tuple of d-ary relation symbols indexed by $\mathsf{T}_R(\Phi)$, say $(R_\mathbf{t})_{\mathbf{t} \in \mathsf{T}_R(\Phi)}$. Unfortunately, the converse implication does not hold in general. For instance, one can check, with the formula φ displayed in (12), that the formula $\exists R_{(x,y,x)} \exists R_{(y,x,y)} \forall x \forall y \tilde{\varphi}$ has a model, while $\exists R \forall x \forall y \varphi$ doesn't have. To get the right-to-left implication in (13), we have to strengthen the hypothesis with some assertion that compels the tuple $(R_\mathbf{t})_{\mathbf{t} \in \mathsf{T}_R(\Phi)}$ to be, in some sense, the d-ary representation of some k-ary relation. All in all, we confront the following question: *Given a σ-structure S, a set $T \subset \mathsf{T}(\Phi)^k$ and a family $(R_\mathbf{t})_{\mathbf{t} \in T}$ of d-ary relations over the domain $[n]$ of S, what are the conditions on $(R_\mathbf{t})_{\mathbf{t} \in T}$ that ensure*

$$\exists R \subseteq [n]^k \textit{ such that } \forall \mathbf{t} \in T, \forall \mathbf{a} \in [n]^d : \; R_\mathbf{t}(\mathbf{a}) = R(\mathbf{t}(\mathbf{a})) \tag{14}$$

Each k-tuple $\mathbf{t} \in T$ defines a function from $[n]^d$ to $[n]^k$, *via* the process of interpretation of terms. (For instance, the triple of terms $\mathbf{t} = (\mathrm{succ}^3 x, x, \mathrm{succ}^2 y)$ maps each couple $(a, b) \in [n]^2$ onto the triple $(a + 3, a, b + 2) \in [n]^3$, where $+$ is the addition modulo n.) Therefore, if in the statement of Fact 5.3 we set:

$$X_i = [n]^d \text{ for each } i, \; Y = [n]^k \text{ and } (f_i)_{i \in I} = (\mathbf{t})_{\mathbf{t} \in T},$$

we get the equivalence of (14) with the following assertion:

$$\forall \mathbf{t}, \mathbf{t}' \in T, \; \forall \mathbf{a}, \mathbf{a}' \in [n]^d : \; \mathbf{t}(\mathbf{a}) = \mathbf{t}'(\mathbf{a}') \Rightarrow R_\mathbf{t}(\mathbf{a}) = R_{\mathbf{t}'}(\mathbf{a}'),$$

which is translated into the logical formula:

$$\bigwedge_{\mathbf{t},\mathbf{t}'\in T} \forall \mathbf{x},\mathbf{x}' :\ \mathbf{t}(\mathbf{x}) = \mathbf{t}'(\mathbf{x}') \to (R_{\mathbf{t}}(\mathbf{x}) \leftrightarrow R_{\mathbf{t}'}(\mathbf{x}'))\,, \tag{15}$$

where $\mathbf{x}$, $\mathbf{x}'$ are d-tuples of first-order variables.

In order to express condition (15) in the required formalism, it remains to reduce the number of quantifiers (remember our logic allows only d universal first-order quantifiers). Since the conjunction and the universal quantification commute, we just have to tackle the case of a single conjunct

$$\forall \mathbf{x},\mathbf{x}' :\ \mathbf{t}(\mathbf{x}) = \mathbf{t}'(\mathbf{x}') \to (R_{\mathbf{t}}(\mathbf{x}) \leftrightarrow R_{\mathbf{t}'}(\mathbf{x}'))\,. \tag{16}$$

To process, we invoke the specificity of coordinate encodings, which has not been mentioned so far in our reasoning. Since succ is the only function symbol in the signature of coordinate structures, all the terms under consideration have the form $\mathrm{succ}^i(u)$ for some $i \in \mathbb{N}$ and some first-order variable u. Hence, the equality $\mathbf{t}(\mathbf{x}) = \mathbf{t}'(\mathbf{x}')$ is a conjunction $\bigwedge_{i\le k}\theta_i$ of k atomic formulas of the type $\mathrm{succ}^a(x) = \mathrm{succ}^b(y)$, with $x \in \mathbf{x}$, $y \in \mathbf{x}'$ and $a, b \ge 0$. Thus the formula displayed in (16) can be written as:

$$\forall \mathbf{x},\mathbf{x}' : (\bigwedge_{i\le k}\theta_i) \to (R_{\mathbf{t}}(\mathbf{x}) \leftrightarrow R_{\mathbf{t}'}(\mathbf{x}')). \tag{17}$$

Since the successor function is cyclic, the assertion $\mathrm{succ}^a(x) = \mathrm{succ}^b(y)$ is equivalent to $x = \mathrm{succ}^{b-a}(y)$ if $a \le b$, and to $y = \mathrm{succ}^{a-b}(x)$ if $a \ge b$. Using this fact, we can eliminate some variables from formula (17). For instance, assume that $\theta_1 \equiv \mathrm{succ}^a(x) = \mathrm{succ}^b(y)$ with $x \in \mathbf{x}$, $y \in \mathbf{x}'$ and $a \le b$. Then, (17) is equivalent to $\forall \mathbf{x},\mathbf{x}'\psi$, where ψ is the formula obtained from (17) by suppressing θ_1 from the conjunction $\bigwedge_{i\le k}\theta_i$ (which therefore becomes $\bigwedge_{2\le i\le k}\theta_i$) and by replacing, in the resulting formula, each occurence of x by the term $\mathrm{succ}^{b-a}(y)$. Notice that the formula thus obtained has the same form as (17). Hence, we can iteratively repeat this procedure until there is no more equality of type $\mathrm{succ}^a(x) = \mathrm{succ}^b(y)$ in the conjunction $\bigwedge\theta_i$ (but it may remain some equalities of the form $\mathrm{succ}^a(x) = \mathrm{succ}^b(x)$, with $x \in \mathbf{x} \cup \mathbf{x}'$).

Example. In order to illustrate this procedure, assume the arity of $R_{\mathbf{t}}$ and $R_{\mathbf{t}'}$ (and hence, that of $\mathbf{x}$ and $\mathbf{x}'$) is 2, and consider the two triples of terms $\mathbf{t} = (\mathrm{succ}^5 x, y, \mathrm{succ}^2 y)$ and $\mathbf{t}' = (\mathrm{succ}^2 y, \mathrm{succ}^3 y, \mathrm{succ}^5 x)$. Suppose we are given the following formula:

$$\Phi \equiv \forall x, y, x', y' :\ \mathbf{t}(x,y) = \mathbf{t}'(x',y') \to (R_{\mathbf{t}}(x,y) \leftrightarrow R_{\mathbf{t}'}(x',y'))$$

in which we want to eliminate as much first-order variables as possible. We first write Φ as:

$$\forall x,y,x',y' : \left\{\begin{array}{ll} \mathrm{succ}^5 x = \mathrm{succ}^2 y' & \wedge \\ \mathrm{succ}^6 y = \mathrm{succ}^2 x' & \wedge \\ \mathrm{succ}^2 x = \mathrm{succ}^5 y' & \end{array}\right\} \to (R_{\mathbf{t}}(x,y) \leftrightarrow R_{\mathbf{t}'}(x',y'))\,.$$

Starting the substitution described above, we suppress the first equality in the conjunction and replace each occurence of y' by $\mathrm{succ}^{5-2}x = \mathrm{succ}^3 x$. Thus we get the equivalent formula:

$$\forall x,y,x',y' : \left\{\begin{array}{ll} \mathrm{succ}^6 y = \mathrm{succ}^2 x' & \wedge \\ \mathrm{succ}^2 x = \mathrm{succ}^8 x & \end{array}\right\} \to \Big(R_{\mathbf{t}}(x,y) \leftrightarrow R_{\mathbf{t}'}(x', \mathrm{succ}^3 x)\Big),$$

An iteration of the process provides the formula:

$$\forall x,y,x',y' : \mathrm{succ}^2 x = \mathrm{succ}^8 x \to \Big(R_{\mathbf{t}}(x,y) \leftrightarrow R_{\mathbf{t}'}(\mathrm{succ}^4 y, \mathrm{succ}^3 x)\Big),$$

which can be more simply writen:

$$\forall x, y, x', y' : x = \mathrm{succ}^6 x \to \Big(R_{\mathbf{t}}(x,y) \leftrightarrow R_{\mathbf{t}'}(\mathrm{succ}^4 y, \mathrm{succ}^3 x)\Big).$$

Afterward, we can obviously get rid of the quantifications over variables that do not appear any more in the matrix of the formula. This finally gives the following formula, equivalent to Φ on coordinate structures:

$$\forall x, y : x = \mathrm{succ}^6 x \to \Big(R_{\mathbf{t}}(x,y) \leftrightarrow R_{\mathbf{t}'}(\mathrm{succ}^4 y, \mathrm{succ}^3 x)\Big).$$

Here, the number of first-order variables agrees with the arity of $R_{\mathbf{t}}$ and $R_{\mathbf{t}'}$. ◁

To conclude, it remains to make the following three remarks:

(1) In the previous example, the number of first-order variables after the elimination procedure equals the arity of $R_{\mathbf{t}}$. This is not a coincidence: In all cases, the procedure provides us with a universal fist-order formula whose number of variable is at most d, the arity of $R_{\mathbf{t}}$ and $R_{\mathbf{t}'}$. This is because the procedure reduces to a single variable each set of variables in $\mathbf{x} \cup \mathbf{x}'$ that are connected by equalities of the form $\mathrm{succ}^a x = \mathrm{succ}^b y$, *via* the initial conjunction $\bigwedge_{i \leq k} \theta_i$. For instance, if this conjunction initially contains the equalities $\mathrm{succ}^2 x = \mathrm{succ}^3 x'$, $\mathrm{succ}^4 x = \mathrm{succ}^2 y'$ and $\mathrm{succ}^3 y = \mathrm{succ}^5 y'$, then the four variables x, y, x', y' will reduce to one during the procedure. More precisely: let us consider the bipartite graph G build on the set of variables $\mathbf{x} \cup \mathbf{x}'$, by linking two variables $u \in \mathbf{x}$, $v' \in \mathbf{x}'$ when there is an equality $\mathrm{succ}^a u = \mathrm{succ}^b v'$ in the conjunction $\bigwedge_{i \leq k} \theta_i$. It is easily seen that the elimination procedure reduces to one variable all variables lying on a same connected component of G. Hence, the number of variables left by the procedure is bounded by the number of connected components of G. Since each connected component contains at least one variable of $\mathbf{x}$ (for G is bipartite between $\mathbf{x}$ and $\mathbf{x}'$), this number of connected components is itself bounded by the cardinality of $\mathbf{x}$, which is d.

(2) Using the strategy described above, each conjunct of (15) can be rewritten with only d universal first-order variables. These rewriting results in a new first-order formula Rep_d^k, equivalent to Condition (15) (which ensures that the relations $R_{\mathbf{t}}$, $\mathbf{t} \in \mathsf{T}_R(\Phi)$, constitute a "$d$-ary representation of a k-ary relation") and has a universal prefix $\forall x_1, \ldots, x_d$ of length d. Clearly, the ESO($\forall^d$, arity k)-formula initially considered in (9), $\Phi \equiv \exists R \forall x_1, \ldots, x_d \, \varphi(\mathbf{x}, R)$, is equivalent on picture encodings to the following ESO($\forall^d$, arity d)-formula:

$$\exists (R_{\mathbf{t}})_{\mathbf{t} \in \mathsf{T}_R(\Phi)} : \mathsf{Rep}_d^k\big((R_{\mathbf{t}})_{\mathbf{t} \in \mathsf{T}_R(\Phi)}\big) \wedge \forall \mathbf{x} \, \tilde{\varphi}(\mathbf{x}, (R_{\mathbf{t}})_{\mathbf{t} \in \mathsf{T}_R(\Phi)}),$$

where $\tilde{\varphi}$ is obtained from φ by replacing each $R(\mathbf{t}(\mathbf{x}))$ by $R_{\mathbf{t}}(\mathbf{x})$.

(3) Finally, we let the reader to verify that this procedure can be extended to any number of relation symbols existentially quantified in Φ. □

Theorem 5.1 immediately proceeds from Propositions 5.2 and 5.4 above. We will remember that:

Over coordinate structures, each ESO(var d)-formula can be written:

$$\Phi \equiv \exists \mathbf{R} \forall \mathbf{x} \bigwedge \bigvee \pm \left\{ \begin{array}{l} \min(\mathrm{succ}^i(x)),\ \max(\mathrm{succ}^i(x)), \\ \mathrm{succ}^i(x) = \mathrm{succ}^j(y), \\ Q_a(\mathrm{succ}^{i_1}(x_{j_1}), \ldots, \mathrm{succ}^{i_p}(x_{j_p})), \\ R(\mathrm{succ}^{i_1}(x_{j_1}), \ldots, \mathrm{succ}^{i_p}(x_{j_p})) \end{array} \right\} \tag{18}$$

where $Q_a \in \sigma$, $R \in \mathbf{R}$ and x, y and the x_i's are all components of $\mathbf{x}$.

Remark 5.5. *Notice that Theorem 5.1 and, more generally, the results of this sections hold with exactly the same proof whatever the arity of the input relation symbols $(Q_s)_{s \in \Sigma}$ is. This won't be the case in some of the forthcoming normalization results, which will heavily depend on the arity of the input structure.*

6. "Sorting" the logic: some motivations and a preliminary example

6.1. Motivations

When introducing Section 5 (see p. 24), we mentioned a particular source of "non-locality" in the logic ESO(var d), due to possible occurences in a given ESO(var d)-formula φ, of both subformulas $R(x, y)$ and $R(y, x)$, for some relation symbol R. Indeed, evaluating such a φ brings about comparing the contents of the two non-adjacent pixels of respective coordinates (x, y) and (y, x). For a cellular automaton, this situation appears as a major obstacle for achieving the evaluation of φ. Of course, this non-local feature persists in ESO($\forall^d$, arity d)-formulas. That's why we now engage in a normalization of ESO($\forall^d$, arity d) into an equivalent logic in which such failure to locality is prevented.

Let us detail our goal. We deal with ESO($\forall^d$, arity d)-formulas conceived to "talk" about $(d-1)$-pictures. Such formulas involve two kinds of relation symbols: the "guessed" ones (those that are existentially quantified) of arity d, and the "input" ones (which are part of the signature), of arity $d-1$. In both cases, we demand that an atomic subformula $R(t_1, \ldots, t_k)$, where k is either d or $d-1$ according to the arity of R and where the t_i's are terms build on the first-order variables $x_1, \ldots, x_d$, fulfill the *sorting condition*: for each $1 \leq i \leq k$, the term t_i is formed with the variable x_i. That is, the variables $x_1, \ldots, x_d$ always occur in the same order in atomic subformulas.

There is a less important cause of non-locality of ESO($\forall^d$, arity d)-formulas, owing to terms that involve iteration of the successor function. For instance, a cellular automaton that attempt to evaluate a formula containing the subformulas $R(x, y)$ and $R(x, \mathrm{succ}^3(y))$, has to check the pixels of coordinate (x, y) and $(x, \mathrm{succ}^3(y))$, which are not adjacent. Actually, this apparent difficulty could be circumvent, since the cellular automaton could carry on this check in constant time. But we choose to include this last characteristic in the objectives of our normalization, because it can be achieved with minor cost, and because it will lighten the proof of Proposition 9.2, which is – we must keep that in mind – the aim of this whole normalization process.

The reader should be now in a position to get the motivation of the technical Definition 6.2 below, which formalizes the requirements that an ESO($\forall^d$, arity d)-formula must fulfill to pretend to "locality". Before coming to that definition, let us introduce a notation that describes pixel which differs from a given d-pixel $\mathbf{x}$ by one unity in one dimension:

Definition 6.1. *Let $n, d > 0$ and $\mathbf{x} = (x_1, \ldots, x_d) \in [n]^d$. For any $i \in \{1, \ldots, d\}$, $\mathbf{x}^{(i)}$ denotes the tuple obtained from $\mathbf{x}$ by replacing its i^{th} component by its own successor. That is:*

$$\mathbf{x}^{(i)} = (x_1, \ldots, x_{i-1}, succ(x_i), x_{i+1}, \ldots, x_d).$$

We now define a *sorted* version of ESO($\forall^d$, arity d). It has very strict requirements on tuples of first-order variables involved in the atoms of the formulas: they must represent pixels that differ by at most one in at most one dimension. That is:

- their components are in the *same order*;
- there is at most *one* occurrence of succ in each atom.

These requirements (and a little more) are formalized below.

Definition 6.2. *Let k, d be two integers such that $d \geq k \geq 1$. A sentence over coordinate structures for k-pictures is in* ESO($\forall^d$, arity d, sorted) *if it is of the form $\exists \mathbf{R} \forall \mathbf{x} \psi(\mathbf{x})$ where*

1. $\mathbf{R}$ *is a list of relation symbols of arity* d;
2. ψ *is a quantifier-free formula whose list of first-order variables is* $\mathbf{x} = (x_1, \dots, x_d)$;
3. *each atom of* ψ *is of one of the following forms:*
   - (i) $Q_s(x_1, \dots, x_k)$, *for* $s \in \Sigma$,
   - (ii) $R(\mathbf{x})$ *or* $R(\mathbf{x}^{(i)})$ *where* $R \in \mathbf{R}$ *and* $i \in [d]$,
   - (iii) $min(x_i)$ *or* $max(x_i)$, *for* $i \in [d]$.

(Remember that beside arithmetic symbols, the tuple of k-ary relations $(Q_s)_{s\in\Sigma}$ *constitutes the core of the signature in the coordinate representation of a k-picture – see Definition 2.3.)*

In the two following sections, we prove the normalization $\mathrm{ESO}(\forall^d, \text{arity}\, d) = \mathrm{ESO}(\forall^d, \text{arity}\, d, \text{sorted}\,)$ for $(d-1)$-pictures (i.e. for $k = d-1$). In Proposition 7.7, we'll deal with conditions (3*ii*) and (3*iii*) of Definition 6.2. At this point, we'll get a normalization of $\mathrm{ESO}(\forall^d, \text{arity}\, d)$ into the so-called "half-sorted logic", denoted by $\mathrm{ESO}(\forall^d, \text{arity}\, d, \text{half-sorted}\,)$. It will remain to manage with the *input* relation symbols. This will be done in Section 8, where we tackle Condition (3*i*).

To facilitate the presentation of the forthcomming results, let's first introduce some notations about tuples and permutations.

Definition 6.3. *Let* $n, d > 0$ *and* $\mathbf{x} \in [n]^d$.

(i) *We denote by* $[\mathbf{x}]_i$ *the* i^{th} *component of* $\mathbf{x}$. *E.g.* $(5,7,2)_2 = 7$.

(ii) *We say that* $\mathbf{x}$ *is* increasing*, and we write* $\mathbf{x}\!\uparrow$, *when* $[\mathbf{x}]_1 \le \dots \le [\mathbf{x}]_d$.

(iii) $\mathcal{S}(d)$ *stands for the set of permutations of* $\{1, \dots, d\}$. *Given pairwise distinct* $\alpha_1, \dots, \alpha_d$ *in* $\{1, \dots, d\}$, *we denote by* $\alpha_1 \dots \alpha_d$ *the permutation* $\alpha \in \mathcal{S}(d)$ *that maps each* i *on* α_i. *Conversely, for* $\alpha \in \mathcal{S}(d)$ *we set* $\alpha_i := \alpha(i)$. *By* $\mathcal{T}(d)$ *we denote the set of transpositions of* $\{1, \dots, d\}$. *Finally, for* $k \le d$ *we write* $\mathcal{I}(k, d)$ *for the set of injections from* $\{1, \dots, k\}$ *into* $\{1, \dots, d\}$.

(iv) *If* $\alpha \in \mathcal{S}(d)$ *and* $\mathbf{x}$ *is a d-tuple, we denote by* $\mathbf{x}_\alpha$ *the d-tuple whose* i^{th} *component is the* α_i^{th} *component of* $\mathbf{x}$. *That is: if* $\mathbf{x} = (x_1, \dots, x_d)$, *then* $\mathbf{x}_\alpha = (x_{\alpha_1}, \dots, x_{\alpha_d})$. *It is less ambiguous to define* $\mathbf{x}_\alpha$ *by the assertion: for any* $i \in \{1, \dots, d\}$,

$$[\mathbf{x}_\alpha]_i = [\mathbf{x}]_{\alpha(i)}.$$

Thus, if β *also belongs to* $\mathcal{S}(d)$, *we get* $[(\mathbf{x}_\alpha)_\beta]_i = [\mathbf{x}_\alpha]_{\beta(i)} = [\mathbf{x}]_{\alpha\beta(i)}$. *Whence the identity:*

$$(\mathbf{x}_\alpha)_\beta = \mathbf{x}_{\alpha\beta}.$$

In particular, $(\mathbf{x}_\alpha)_{\alpha^{-1}} = \mathbf{x}$.

(v) *For* $\alpha \in \mathcal{S}(d)$ *and* $n > 0$, *we set* $[\alpha] = \{\mathbf{x} \in [n]^d \text{ s.t. } \mathbf{x}_\alpha\!\uparrow\}$. *Clearly,* $[n]^d = \bigcup_{\alpha\in\mathcal{S}(d)} [\alpha]$. *Beside, denoting by* id *the identity on* $\{1, \dots, d\}$, *we get* $\mathbf{x} \in [\mathsf{id}]$ *iff* $\mathbf{x}\!\uparrow$ *and therefore*

$$\mathbf{x} \in [\alpha] \textit{ iff } \mathbf{x}_\alpha \in [\mathsf{id}].$$

In coherence with Definition 6.1, the arrangement of $\mathbf{x}^{(i)}$ *according to the permutation* α *is denoted by* $(\mathbf{x}^{(i)})_\alpha$.

Example. Take $\mathbf{x} = (5,3,7,2)$ in $[9]^4$ and $\alpha = 4213$, $\beta = 1432$ in $\mathcal{S}(4)$. Then $\mathbf{x}_\alpha = (2,3,5,7)$ is increasing while $\mathbf{x}_\beta = (5,2,7,3)$ is not. Besides, $\mathbf{x}^{(3)} = (5,3,8,2)$ while $(\mathbf{x}_\alpha)^{(3)} = (2,3,6,7)$ and $(\mathbf{x}^{(3)})_\alpha = (2,3,5,8) = (\mathbf{x}_\alpha)^{(4)}$. ◃

The two upcoming sections are devoted to the normalization $\mathrm{ESO}(\mathrm{var}\, d) = \mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \mathrm{sorted})$ on coordinate structures of dimension $d-1$. This result is stated in Theorem 8.7. Because of the previous steps of normalization (Propositions 5.2 and 5.4), it amounts to proving that $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d) = \mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \mathrm{sorted})$ on $(d-1)$-coordinate structures. Remember this essentially means that R-atoms of an $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$-formula can be compelled to fit the form $R(x_1, \dots, x_k)$ if R is in the signature, or $R(\mathbf{x})$ (resp. $R(\mathbf{x}^{(i)})$) if R is existentially quantified (see Definition 6.2). It will appear that the technics involved in this normalizations of R-atoms are quite different, depending on whether R is part of the input or not. Hence, we'll prove this result in two steps, matching the two above mentionned cases:

In Section 7, we state a weaker form of the inclusion $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d) \subseteq \mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \mathrm{sorted})$. There, we relax one of the constraints on the target logic. Namely, we replace constraint (3*i*) of Definition 6.2 by the more liberal requisit that each Q_s-atom has the form $Q_s(x_{\iota_1}, \dots, x_{\iota_k})$, where ι is an injection from $[k]$ into $[d]$. This gives rise to the definition of the so-called *half-sorted logic*:

Definition 6.4. *Let* k, d *be two integers such that* $d \geq k \geq 1$. *A sentence over coordinate structures for* k*-pictures is in* $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \mathrm{half\text{-}sorted})$ *if it is of the form* $\exists \mathbf{R} \forall \mathbf{x} \psi(\mathbf{x})$ *where*

1. $\mathbf{R}$ *is a list of relation symbols of arity* d*;*
2. ψ *is a quantifier-free formula whose list of first-order variables is* $\mathbf{x} = (x_1, \dots, x_d)$*;*
3. *each atom of* ψ *is of one of the following forms:*
   (*a*) $Q_s(\mathbf{x}_\iota)$*, where* $s \in \Sigma$ *and* ι *is an injection from* $[k]$ *into* $[d]$*,*
   (*b*) $R(\mathbf{x})$ *or* $R(\mathbf{x}^{(i)})$ *where* $R \in \mathbf{R}$ *and* $i \in [d]$*,*
   (*c*) *min*(x_i) *or max*(x_i)*, for* $i \in [d]$.

Proposition 7.7 state the equality

$$\mathrm{ESO}(\forall^d, \mathrm{arity}\, d) = \mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \mathrm{half\text{-}sorted})$$

on coordinate pictures while Proposition 8.6 completes the process of normalization by stating the equality

$$\mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \mathrm{half\text{-}sorted}) = \mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \mathrm{sorted})$$

on coordinate pictures of dimension $d-1$. We chose in both case to introduce the proofs with a simple example illustrating some important features.

6.2. An example

Let's now illustrate the first step of the normalization. We want to prove that each formula Φ in $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$ can be written in $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \mathrm{half\text{-}sorted})$ on coordinate structures. Assume $d = 2$ and Φ is a formula in $\mathrm{ESO}(\forall^2, \mathrm{arity}\, 2)$ that has the simple form $\Phi = \exists R \forall x \forall y \psi(x, y)$, where R is a binary relation symbol and ψ is a quantifier-free formula in which the R-atoms have the following forms:

(1) $R(x, y)$; (2) $R(\mathrm{succ}(x), y)$; (3) $R(x, \mathrm{succ}(y))$; (4) $R(y, x)$.

	$R(x,y)$	$R(\mathrm{succ}(x),y)$	$R(x,\mathrm{succ}(y))$	$R(y,x)$
$\psi_<(x,y)$	$R_1(x,y)$	$R_1(\mathrm{succ}(x),y)$	$R_1(x,\mathrm{succ}(y))$	$R_2(x,y)$
$\psi_=(x,y)$	$R_1(x,y)$	$R_2(x,\mathrm{succ}(y))$	$R_1(x,\mathrm{succ}(y))$	$R_1(x,y)$
$\psi_>(x,y)$	$R_2(y,x)$	$R_2(y,\mathrm{succ}(x))$	$R_2(\mathrm{succ}(y),x)$	$R_1(y,x)$

Table 1: Replacement of R-atoms by R_1- or R_2-atoms

(We don't evoke the form of others atoms since we focus here on the treatment of the guessed relation R.) For given x and y, the pixels (x,y), $(\mathrm{succ}(x),y)$ and $(x,\mathrm{succ}(y))$ are adjacent, while (y,x) is arbitrarily far from the former, since it is symmetric of (x,y) with respect to the diagonal $x=y$. Hence we have to eliminate subformulas ot type (4). The intuitive idea is to "fold" the picture along this diagonal. Then R is represented by two "half relations" R_1 and R_2, that are superposed in the half square $x \leq y$ above the diagonal.

Thus, R_1 and R_2 are binary relations whose intuitive meaning is the following: for points (x,y) such that $x \leq y$, one has $R_1(x,y) = R(x,y)$ and $R_2(x,y) = R(y,x)$. By this transformation, both informations $R(x,y)$ and $R(y,x)$, dealing with symmetric pixels (x,y) and (y,x), are accessible by checking the validity of the assertions $R_1(x,y)$ and $R_2(x,y)$ on the sole pixel (x,y), where $x \leq y$. The case $y \leq x$ is similar. This solves the problem of neighborhood.

Formaly, the sentence $\Phi = \exists R \forall x \forall y \psi(x,y)$ is normalized as follows. Let $\mathsf{coherent}(x,y)$ denote the formula $x = y \rightarrow (R_1(x,y) \leftrightarrow R_2(x,y))$ whose universal closure ensures the coherence of R_1 and R_2 on the common part of R they both represent, that is the diagonal $x=y$. Using R_1 and R_2, it is not difficult to construct a formula

$$\psi'(x,y) = \mathsf{coherent}(x,y) \wedge \begin{pmatrix} x<y \rightarrow \psi_<(x,y) & \wedge \\ x=y \rightarrow \psi_=(x,y) & \wedge \\ x>y \rightarrow \psi_>(x,y) & \end{pmatrix}$$

such that the sentence $\Phi' = \exists R_1, R_2 \forall x, y \psi'(x,y)$, which belongs to $\mathrm{ESO}(\forall^2, \mathrm{arity}\,2)$, is equivalent to Φ. Let us describe and justify its precise form and meaning.

The formulas $\psi_<(x,y)$, $\psi_=(x,y)$ and $\psi_>(x,y)$ are obtained from formula $\psi(x,y)$ by substitution of R_1- or R_2-atoms for R-atoms, according to the cases described in Table 1. It is easy to check that each replacement is correct according to its case. For instance, it is justified to replace each atom of the form $R(x,\mathrm{succ}(y))$ in ψ by $R_2(\mathrm{succ}(y),x)$ when $x>y$ (in order to obtain the formula $\psi_>(x,y)$), because when $x>y$ we get $\mathrm{succ}(y) \leq x$ and hence the equivalence $R(x,\mathrm{succ}(y)) = R_2(\mathrm{succ}(y),x)$ holds, by definition of R_2.

Notice that the variables x,y always occur in this order in each R_1- or R_2-atom of the formulas $\psi_<$ and $\psi_=$ (see Table 1). At the opposite, they always occur in the reverse order y,x in the formula $\psi_>(x,y)$. This is not a problem because, by symmetry, the roles of x and y can be exchanged and the universal closure $\forall x,y : x>y \rightarrow \psi_>(x,y)$ is trivially equivalent to $\forall x,y : y>x \rightarrow \psi_>(y,x)$. So, the above sentence Φ' – and hence, the original sentence Φ – is equivalent to the sentence denoted Φ" obtained by replacing in Φ' the subformula $x>y \rightarrow \psi_>(x,y)$ by $y>x \rightarrow \psi_>(y,x)$. By construction, relation symbols R_1 and R_2 only occur in Φ within atoms of the three required "sorted" forms: $R_i(x,y)$, $R_i(\mathrm{succ}(x),y)$ or $R_i(x,\mathrm{succ}(y))$.

Finally, notice that the resulting sentence involves equalities and inequalities although it shoud not be the case, according to Definition 6.4. We will see how to fix this point in the next section, when dealing with the general case.

7. Sorting guessed relations

For the general case, the steps of the proof are similar to those presented above, but the notations and details of the proof are more involved. Let us succinctly describe the ESO relations of arity d to be introduced in the main normalization step, and corresponding to the relations R_1 and R_2 defined in Subsection 6.2. Here again, each ESO relation symbol R of the original sentence Φ in $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$ is replaced by – or, intuitively, "divided into" – $d!$ new ESO relation symbols R_α of the same arity d. Here, α ranges over permutations of the set of indices $[d]$. The intended meaning of each relation R_α is the following: for each tuple $(a_1, \ldots, a_d) \in [n]^d$ such that $a_1 \le a_2 \ldots \le a_d$, we have

$$R_\alpha(a_1, \ldots, a_d) = R(a_{\alpha(1)}, \ldots, a_{\alpha(d)}).$$

Then, we introduce a partition of the domain $[n]^d$ into subdomains, similar to the partition of $[n]^2$ into the three subsets of equation $x < y$, $x = y$, $x > y$, described in Subsection 6.2. According to the case (i.e., to the subdomain of the partition), this allows to replace each R-atom in Φ by an atom of the sorted form $R_\alpha(\mathbf{x})$ or $R_\alpha(\mathbf{x}^{(i)})$, where $\mathbf{x} = (x_1, \ldots, x_d)$ and $1 \le i \le d$ (remember $\mathbf{x}^{(i)}$ is the tuple $\mathbf{x}$ where x_i is replaced by $\mathrm{succ}(x_i)$ – see Definition 6.1). Finally, the equalities are eliminated in order to obtain a sentence that fully fit the syntactical restrictions of $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \text{half-sorted})$.

Fact 7.1. *On coordinate structures, any formula* $\Phi = \exists \mathbf{R} \forall \mathbf{x} \varphi(\mathbf{x}, \mathbf{R}) \in \mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$ *can be written in such a way that:*

(a) *Each atomic subformula* $P(t_1, \ldots, t_p)$ *of* φ*, where* P *is either an input or a guessed relation symbol, fulfills* $Var(t_i) \cap Var(t_j) = \emptyset$ *for every* $1 \le i < j \le p$.

(b) *Each guessed relation symbol* $R \in \mathbf{R}$ *has arity* d *exactly.*

Proof. The proof of (a) is quite immediate and we just illustrate it with an example. Assume that Φ involves the P-atom $P(\mathrm{succ}^2 x_1, x_2, \mathrm{succ}\, x_1, \mathrm{succ}^3 x_2)$ for some relation symbol P. Then clearly Φ is equivalent, on picture-structures, to:

$$\exists \mathbf{R} \exists A \forall \mathbf{x} : \tilde{\varphi}(\mathbf{x}, \mathbf{R}, A)\ \wedge (x_1 = \mathrm{succ}\, x_3 \wedge x_4 = \mathrm{succ}^3 x_2) \to (A(x_2, x_3) \leftrightarrow P(x_1, x_2, x_3, x_4))$$

where $\tilde{\varphi}$ is obtained from φ by substituting the formula $A(x_2, \mathrm{succ}(x_1))$ for each occurrence of the atom $P(\mathrm{succ}^2 x_1, x_2, \mathrm{succ}\, x_1, \mathrm{succ}^3 x_2)$.

In order to prove (b), assume for simplicity that $\mathbf{R}$ reduces to a single relational symbol R of arity $p < d$. The idea is to replace R by a d-ary relation symbol R' with $d - p$ dummy arguments. Clearly, Φ is equivalent to the formula:

$$\exists R' \forall \mathbf{x} :\ \tilde{\varphi} \wedge \bigwedge_{p < i \le d} R'(x_1, \ldots, x_i, \ldots, x_d) \leftrightarrow R'(x_1, \ldots, \mathrm{succ}(x_i), \ldots, x_d).$$

Here, $\tilde{\varphi}$ is obtained from φ by replacing each atomic subformula $R(t_1, \ldots, t_p)$ by $R'(t_1, \ldots, t_p, x_{i_1}, \ldots, x_{i_{d-p}})$, where $x_{i_1}, \ldots, x_{i_{d-p}}$ is the complete list of distinct variables among $\mathbf{x}$ that do not occur in $t_1, \ldots, t_p$. ◻

Remark 7.2. *Considering this proof, we can sharpen the statement of Fact 7.1 by adding the following requirement: if the* Q_s*'s have arity* $k \le d$*,*

(c) *Each* Q_s*-atom of* Φ*, for* $s \in \Sigma$*, has the form* $Q_s(x_{\iota_1}, \ldots, x_{\iota_{d-1}})$ *where* $\iota \in \mathcal{I}(k, d)$.

(Remember that $\mathcal{I}(k,d)$ is the set of injections from $\{1,\dots,k\}$ into $\{1,\dots,d\}$ – see Definition 6.3(iii).)

Fact 7.3. *On coordinate structures, any formula $\Phi = \exists \mathbf{R} \forall \mathbf{x} \varphi(\mathbf{x},\mathbf{R}) \in \mathrm{ESO}(\forall^d, \text{arity } d)$ can be written in such a way that each R-atom of φ, with $R \in \mathbf{R}$, has one of the two forms: $R(\mathbf{x}^{(i)})$ or $R(\mathbf{x}_\alpha)$, where $\alpha \in \mathcal{S}(d)$.*

Proof. We prove the result for $d = 3$. The general case is similar. Let ℓ be the maximal value of an $i \in \mathbb{N}$ such that $\mathrm{succ}^i(x)$ occurs in Φ, for any $x \in \mathbf{x}$. For each $R \in \mathbf{R}$, we introduce new d-ary relation symbols $R_{i,j,k}$ for every $i, j, k \leq \ell$. We want to force the following interpretations of the $R_{i,j,k}$'s:

$$R_{i,j,k}(u_1,u_2,u_3) = R(\mathrm{succ}^i u_1, \mathrm{succ}^j u_2, \mathrm{succ}^k u_3).$$

This is done inductively, with the formulas:

- $\forall \mathbf{x} : R_{0,0,0}(x_1,x_2,x_3) \leftrightarrow R(x_1,x_2,x_3)$
- $\forall \mathbf{x} : \bigwedge_{i<\ell} \bigwedge_{j,k\leq\ell} (R_{i+1,j,k}(x_1,x_2,x_3) \leftrightarrow R_{i,j,k}(\mathrm{succ}(x_1),x_2,x_3))$
- $\forall \mathbf{x} : \bigwedge_{j<\ell} \bigwedge_{i,k\leq\ell} (R_{i,j+1,k}(x_1,x_2,x_3) \leftrightarrow R_{i,j,k}(x_1,\mathrm{succ}(x_2),x_3))$
- $\forall \mathbf{x} : \bigwedge_{k<\ell} \bigwedge_{i,j\leq\ell} (R_{i,j,k+1}(x_1,x_2,x_3) \leftrightarrow R_{i,j,k}(x_1,x_2,\mathrm{succ}(x_3)))$

Factorizing the quantifications and using notations of Definition 6.3, the conjunction of these formulas can be written:

$$\forall \mathbf{x} \left\{ \begin{array}{l} R_{0,0,0}(\mathbf{x}) \leftrightarrow R(\mathbf{x}) \wedge \\ \bigwedge_{i<\ell} \bigwedge_{j,k\leq\ell} (R_{i+1,j,k}(\mathbf{x}) \leftrightarrow R_{i,j,k}(\mathbf{x}^{(1)})) \wedge \\ \bigwedge_{j<\ell} \bigwedge_{i,k\leq\ell} (R_{i,j+1,k}(\mathbf{x}) \leftrightarrow R_{i,j,k}(\mathbf{x}^{(2)})) \wedge \\ \bigwedge_{k<\ell} \bigwedge_{i,j\leq\ell} (R_{i,j,k+1}(\mathbf{x}) \leftrightarrow R_{i,j,k}(\mathbf{x}^{(3)})) \end{array} \right\}$$

Let us denotes by $\mathsf{decomp}(R,(R_{i,j,k})_{i,j,k\leq\ell})$ this last formula. It clearly fulfills the condition of the statement. Now, consider the formula

$$\exists \mathbf{R} \exists ((R_{i,j,k})_{i,j,k\leq\ell})_{R\in\mathbf{R}} : \bigwedge_{R\in\mathbf{R}} \mathsf{decomp}(R,(R_{i,j,k})_{i,j,k\leq\ell}) \ \wedge \ \forall \mathbf{x} \tilde{\varphi}, \tag{19}$$

where $\tilde{\varphi}$ is obtained from φ by the substitutions:

$$R(\mathrm{succ}^i x, \mathrm{succ}^j y, \mathrm{succ}^k z) \rightsquigarrow R_{i,j,k}(x,y,z).$$

Then, the formula (19) is equivalent to Φ and also fits the requirements of Fact 7.3. It is the rewriting of Φ announced. □

Notice that in the so-obtained formula, one can assume that each equality has the form $x = \mathrm{succ}^i y$: it suffices to replace each general equality $\mathrm{succ}^i(x) = \mathrm{succ}^j(y)$ by $\mathrm{succ}^{i-j}(x) = y$ or $x = \mathrm{succ}^{j-i}(y)$ according to the sign of $i - j$. Thus, Fact 7.1, Remark 7.2 and Fact 7.3 finally result in: On coordinate structures of dimension $k \le d$, any formula $\Phi \in \mathrm{ESO}(\forall^d, \text{arity } d)$ can be written:

$$\Phi \equiv \exists \mathbf{R} \forall \mathbf{x} \bigwedge \bigvee \pm \left\{ \begin{array}{l} \min(\mathrm{succ}^i(x)),\ \max(\mathrm{succ}^i(x)), \\ x = \mathrm{succ}^i(y), \\ Q_s(\mathbf{x}_\iota),\ R(\mathbf{x}_\alpha),\ R(\mathbf{x}^{(i)}) \end{array} \right\} \tag{20}$$

where $s \in \Sigma$, $R \in \mathbf{R}$, $x, y \in \mathbf{x}$, $\iota \in \mathcal{I}(k, d)$, $\alpha \in \mathcal{S}(d)$ and $i \in \{1, \ldots, d\}$.

Toward a last refinement, we can use the trick of the proof of Fact 7.3 to further simplify atomic formulas involving the predicate symbols $=$, min and max. In order to deal with $=$, for instance, we existentially quantify over new binary relation symbols $S_1, \ldots, S_\ell$, where ℓ is the maximal integer such that an equality of the form $x = \mathrm{succ}^\ell(y)$ occurs in the formula. We force these symbols to fit the interpretation: $S_i(x, y)$ iff $x = \mathrm{succ}^i(y)$, with the formula:

$$(S_0(x, y) \leftrightarrow x = y) \wedge \bigwedge_{0 < i \le \ell} (S_i(x, y) \leftrightarrow S_{i-1}(x, \mathrm{succ}(y)))$$

in which the sole equality has the form $x = y$. Hence, substituting $S_i(x, y)$ for each subformula $x = \mathrm{succ}^i(y)$ of Φ, we ensure that all equalities in Φ have the form $x = y$ for some $x, y \in \mathbf{x}$. At this step, one can replace the binary symbols $S_1, \ldots, S_\ell$ by d-ary relation symbols, as in the proof of Fact 7.1(*b*), without introducing new equalities. The same process can be carry on to simplify min- and max-atoms. Finally, Equation (20) can be more precisely formulated:

> On coordinate structures of dimension $k \le d$, any formula $\Phi \in \mathrm{ESO}(\forall^d, \text{arity } d)$ can be written:
>
> $$\Phi \equiv \exists \mathbf{R} \forall \mathbf{x} \bigwedge \bigvee \pm \left\{ \begin{array}{l} \min(x),\ \max(x), x = y, \\ Q_s(\mathbf{x}_\iota),\ R(\mathbf{x}_\alpha),\ R(\mathbf{x}^{(i)}) \end{array} \right\} \tag{21}$$
>
> where $s \in \Sigma$, $R \in \mathbf{R}$, $x, y \in \mathbf{x}$, $\iota \in \mathcal{I}(k, d)$, $\alpha \in \mathcal{S}(d)$ and $i \in \{1, \ldots, d\}$.

(Notice that despite the appearances, terms involving the function symbol succ have not vanished: they are hidden in the notation $R(\mathbf{x}^{(i)})$.)

It remains to prove that we can get rid of the atomic formulas $R(\mathbf{x}_\alpha)$, where $\alpha \ne \mathsf{id}$. This part is rather technical, so we provide some preliminary explanations before stating the logical framework which allows the normalization. In order to get rid of each literal of the form $R(x_\alpha)$, we will divide the set $R \subseteq [n]^d$ in $d!$ relations $R_\alpha \subseteq [n]^d$, each corresponding to a given permutation α of $\{1, \ldots, d\}$.

Definition 7.4. *For $R \subseteq [n]^d$ and for each $\alpha \in \mathcal{S}(d)$, we define a d-ary relation R_α on $[n]$ by:*

$$R_\alpha = \{\mathbf{x} \in [\mathsf{id}] \text{ s.t. } R(\mathbf{x}_{\alpha^{-1}})\}.$$

Alternatively, R_α can be defined by: $R_\alpha = \{\mathbf{x}_\alpha : \mathbf{x} \in R \cap [\alpha]\}$. (Refer to Definition 6.3(v) for the meaning of notations $[\alpha]$ and $[\mathsf{id}]$.)

Thus, Definition 7.4 associates with each $R \subseteq [n]^d$ a family $(R_\alpha)_{\alpha \in \mathcal{S}(d)}$ of relations, each of which is entirely contained in the set $[\mathsf{id}]$. This family is intended to represent R through its $d!$ fragments according to the partition $[n]^d = \bigcup_{\alpha \in \mathcal{S}(d)} [\alpha]$. Namely, each R_α encodes the fragment $R \cap [\alpha]$ over $[\mathsf{id}]$.

Actually, $\bigcup_{\alpha \in \mathcal{S}(d)}[\alpha]$ is not really a partition, since the $[\alpha]$'s can overlap. Hence, Definition 7.4 induces some connexions between the relations R_α: if some $\mathbf{x}$ is both in $[\alpha]$ and in $[\beta]$, or equivalently, if $\mathbf{x}_\alpha = \mathbf{x}_\beta$, then $\mathbf{x} \in R \cap [\alpha]$ iff $\mathbf{x} \in R \cap [\beta]$ and hence, by Definition 7.4: $R_\alpha(\mathbf{x}_\alpha) = R_\beta(\mathbf{x}_\beta)$. We will keep in mind :

$$\forall \alpha, \beta \in \mathcal{S}(d), \forall \mathbf{x} \in [n]^d : \mathbf{x}_\alpha = \mathbf{x}_\beta \Rightarrow R_\alpha(\mathbf{x}_\alpha) = R_\beta(\mathbf{x}_\beta). \tag{22}$$

The following lemma states that condition (22) ensures that the R_α's issue from a single relation R, according to Definition 7.4. Besides, a new formulation of the condition is given in Item 3 of the lemma, that will better fit our syntactical restrictions.

Lemma 7.5. *Let $(R_\alpha)_{\alpha \in \mathcal{S}(d)}$ be a family of d-ary relations on $[n]$ such that $R_\alpha \subseteq [\mathsf{id}]$ for each α. The following are equivalent:*

1. *$\exists R \subseteq [n]^d$ such that for each $\alpha \in \mathcal{S}(d)$: $R_\alpha = \{\mathbf{x} \in [\mathsf{id}]$ s.t. $R(\mathbf{x}_{\alpha^{-1}})\}$;*
2. *$\forall \alpha, \beta \in \mathcal{S}(d)$, $\forall \mathbf{x} \in [n]^d$: $\mathbf{x}_\alpha = \mathbf{x}_\beta \Rightarrow R_\alpha(\mathbf{x}_\alpha) = R_\beta(\mathbf{x}_\beta)$;*
3. *$\forall \alpha \in \mathcal{S}(d)$, $\forall \tau \in \mathcal{T}(d)$, $\forall \mathbf{x} \in [n]^d$: $\mathbf{x} = \mathbf{x}_\tau \Rightarrow R_\alpha(\mathbf{x}) = R_{\alpha\tau}(\mathbf{x})$.*

(Remember that $\mathcal{T}(d)$ denotes the set of transpositions of $\{1, \ldots, d\}$ – see Definition 6.3(iii).)

Proof. $1 \Rightarrow 2$: See Equation (22).

$2 \Rightarrow 1$: For $(R_\alpha)_{\alpha \in \mathcal{S}(d)}$ fulfilling 2, consider the relation $R \subseteq [n]^d$ defined by:

$$R(\mathbf{x}) \text{ iff } R_\alpha(\mathbf{x}_\alpha) \text{ for some } \alpha \text{ such that } \mathbf{x}_\alpha\uparrow . \tag{23}$$

This definition is well formed, since for any $\alpha, \beta \in \mathcal{S}(d)$ and any $\mathbf{x} \in [n]^d$ such that both $\mathbf{x}_\alpha\uparrow$ and $\mathbf{x}_\beta\uparrow$ hold, we have $\mathbf{x}_\alpha = \mathbf{x}_\beta$ and thus, by 2, $R_\alpha(\mathbf{x}_\alpha) = R_\beta(\mathbf{x}_\beta)$. Now, let $\alpha \in \mathcal{S}(d)$. For any $\mathbf{x} \in [\mathsf{id}]$ we have $(\mathbf{x}_{\alpha^{-1}})_\alpha\uparrow$ (since $(\mathbf{x}_{\alpha^{-1}})_\alpha = \mathbf{x}$) and hence, by (23), $R_\alpha(\mathbf{x}) = R_\alpha((\mathbf{x}_{\alpha^{-1}})_\alpha) = R(\mathbf{x}_{\alpha^{-1}})$. Besides, $R_\alpha(\mathbf{x}) = 0$ for any $\mathbf{x} \notin [\mathsf{id}]$, since $R_\alpha \subseteq [\mathsf{id}]$. Thus R_α is obtained from R as required in 1.

$2 \Rightarrow 3$: Let $\alpha \in \mathcal{S}(d)$, $\tau \in \mathcal{T}(d)$ and $\mathbf{x} \in [n]^d$ such that $\mathbf{x} = \mathbf{x}_\tau$. Set $\mathbf{y} = \mathbf{x}_{\alpha^{-1}}$. Then, $\mathbf{y}_\alpha = \mathbf{x} = \mathbf{x}_\tau = (\mathbf{y}_\alpha)_\tau = \mathbf{y}_{\alpha\tau}$. Therefore we get by 2: $R_\alpha(\mathbf{y}_\alpha) = R_{\alpha\tau}(\mathbf{y}_{\alpha\tau})$, and hence: $R_\alpha(\mathbf{x}) = R_{\alpha\tau}(\mathbf{x})$.

$3 \Rightarrow 2$: Let $\alpha, \beta \in \mathcal{S}(d)$ and $\mathbf{x} \in [n]^d$ such that $\mathbf{x}_\alpha = \mathbf{x}_\beta$. For $\mathbf{y} = \mathbf{x}_\alpha$, the equality $\mathbf{x}_\alpha = \mathbf{x}_\beta$ can be written $\mathbf{y} = \mathbf{y}_{\alpha^{-1}\beta}$. It means that the permutation $\alpha^{-1}\beta$ exchanges integers that index equal components of $\mathbf{y}$. It is easily seen that this property can be required for each transposition occuring in a decomposition of $\alpha^{-1}\beta$ on $\mathcal{T}(d)$. That is, there exist some transpositions $\tau_1, \ldots, \tau_k \in \mathcal{T}(d)$ such that $\alpha^{-1}\beta = \tau_1 \ldots \tau_k$ and $\mathbf{y} = \mathbf{y}_{\tau_1} = \mathbf{y}_{\tau_1\tau_2} = \cdots = \mathbf{y}_{\tau_1 \ldots \tau_k}$. Then, applying 3 to these successive tuples, we get: $R_\alpha(\mathbf{y}) = R_{\alpha\tau_1}(\mathbf{y}_{\tau_1}) = R_{\alpha\tau_1\tau_2}(\mathbf{y}_{\tau_1\tau_2}) = \cdots = R_{\alpha\tau_1 \ldots \tau_k}(\mathbf{y}_{\tau_1 \ldots \tau_k})$. Hence $R_\alpha(\mathbf{y}) = R_\beta(\mathbf{y}_{\alpha^{-1}\beta})$, that is $R_\alpha(\mathbf{x}_\alpha) = R_\beta(\mathbf{x}_\beta)$, as required. □

Lemma 7.6. *Let R and $(R_\alpha)_{\alpha \in \mathcal{S}(d)}$ be defined as in Definition 7.4. Let $\alpha, \beta \in \mathcal{S}(d)$ and $i \in \{1, \ldots, d\}$. For any $\mathbf{x} \in [\alpha]$:*

1. *$R(\mathbf{x}_\beta) = R_{\beta^{-1}\alpha}(\mathbf{x}_\alpha)$.*
2. *$R(\mathbf{x}^{(i)})$ is equivalent to:*

$$\left(x_i = x_{\alpha_d} \wedge R_{\alpha\tau_{j,d}}((x_\alpha)^{(d)})\right) \vee \bigvee_{i \leq k < d} \left(x_i = x_{\alpha_k} \neq x_{\alpha_{k+1}} \wedge R_{\alpha\tau_{j,k}}((x_\alpha)^{(k)})\right),$$

where $j = \alpha^{-1}(i)$. (Here, $\tau_{u,v}$ denotes the transposition swapping u and v.)

Proof. **1.** Since $\mathbf{x} \in [\alpha]$, and hence $\mathbf{x}_\alpha \in [\mathsf{id}]$, Definition 7.4 yields:

$$R_{\beta^{-1}\alpha}(\mathbf{x}_\alpha) = R((\mathbf{x}_\alpha)_{(\beta^{-1}\alpha)^{-1}}) = R((\mathbf{x}_\alpha)_{\alpha^{-1}\beta}) = R(\mathbf{x}_{\alpha\alpha^{-1}\beta}) = R(\mathbf{x}_\beta).$$

2. From $\mathbf{x}^{(i)} = (x_1, \ldots, \mathrm{succ}(x_i), \ldots, x_d)$ we get: $(\mathbf{x}^{(i)})_\alpha = (x_{\alpha_1}, \ldots, x_{\alpha_{j-1}}, \mathrm{succ}(x_i), x_{\alpha_{j+1}}, \ldots, x_{\alpha_d})$, where $j = \alpha^{-1}(i)$. Since $x_{\alpha_1} \leq \cdots \leq x_{\alpha_d}$, the above tuple $(\mathbf{x}^{(i)})_\alpha$ is almost increasingly ordered. More precisely, there exists $k \in \{1, \ldots, d\}$ such that:

$$x_{\alpha_1} \leq \cdots \leq x_{\alpha_{j-1}} \leq x_i = x_{\alpha_{j+1}} = \cdots = x_{\alpha_k} \leq x_{\alpha_{k+1}} \leq \cdots \leq x_{\alpha_d},$$

where $j = \alpha^{-1}(i)$. Clearly, the largest such k is characterized by: $(k = d)$ or $(x_i = x_{\alpha_k} < x_{\alpha_{k+1}})$. Or equivalently, by:

$$(x_i = x_{\alpha_d}) \text{ or } (x_i = x_{\alpha_k} \neq x_{\alpha_{k+1}}). \tag{24}$$

If we denote by $\tau_{j,k}$ the transposition over $\{1, \ldots, d\}$ which permutes j and k, the definition of k yields that the tuple

$$(x^{(i)})_{\alpha\tau_{j,k}} = (x_{\alpha_1}, \ldots, x_{\alpha_{j-1}}, \boxed{x_{\alpha_k}}, x_{\alpha_{j+1}}, \ldots, x_{\alpha_{k-1}}, \boxed{\mathrm{succ}(x_i)}, x_{\alpha_{k+1}}, \ldots, x_{\alpha_d})$$

is increasing. Hence, $R(\mathbf{x}^{(i)}) = R_{\alpha\tau_{j,k}}((x^{(i)})_{\alpha\tau_{j,k}})$. Besides, since $x_{\alpha_k} = x_i$, the tuple $(x^{(i)})_{\alpha\tau_{j,k}}$ above can also be written:

$$(x^{(i)})_{\alpha\tau_{j,k}} = (x_{\alpha_1}, \ldots, x_{\alpha_{j-1}}, \boxed{x_i}, x_{\alpha_{j+1}}, \ldots, x_{\alpha_{k-1}}, \boxed{\mathrm{succ}(x_{\alpha_k})}, x_{\alpha_{k+1}}, \ldots, x_{\alpha_d}).$$

That is: $(x^{(i)})_{\alpha\tau_{j,k}} = (x_\alpha)^{(k)}$. Therefore: $R(\mathbf{x}^{(i)}) = R_{\alpha\tau_{j,k}}((x_\alpha)^{(k)})$. We can finally state: there exists a sole $k \in \{i, \ldots, d\}$ defined by (24), and for this k we have: $R(\mathbf{x}^{(i)}) = R_{\alpha\tau_{j,k}}((x_\alpha)^{(k)})$. Reminding that $j = \alpha^{-1}(i)$, the conclusion immediatly proceeds. ◻

Proposition 7.7. *For $d > 1$, $\mathrm{ESO}(\forall^d, \text{arity } d) \subseteq \mathrm{ESO}(\forall^d, \text{arity } d, \text{half-sorted})$ on coordinate structures of dimension $d - 1$.*

Proof. To simplify, assume we want to translate in $\mathrm{ESO}(\forall^d, \text{arity } d, \text{half-sorted})$ some $\mathrm{ESO}(\forall^d, \text{arity } d)$-formula of the very simple shape: $\Phi \equiv \exists R \forall \mathbf{x} \varphi(\mathbf{x}, R)$, where R is a (single) d-ary relation symbol, $\mathbf{x}$ is a d-tuple of first-order variables, and φ is a quantifier-free formula. Since the sets $[\alpha]$, $\alpha \in \mathcal{S}(d)$, cover the domain $[n]$, we obtain an equivalent rewriting of Φ with the following artificial relativization:

$$\Phi \equiv \exists R \forall \mathbf{x} \bigwedge_{\alpha \in \mathcal{S}(d)} (\mathbf{x} \in [\alpha] \to \varphi).$$

Furthermore, all R-atoms of φ can be assumed of the form $R(\mathbf{x}_\alpha)$ or $R(\mathbf{x}^{(i)})$, thanks to Fact 7.3. To get rid of the literals $R(\mathbf{x}_\alpha)$, we substitute to R a tuple of relations $(R_\alpha)_{\alpha \in \mathcal{S}(d)}$ that encode R on the sets $[\alpha]$. We proved in Lemma 7.5 that this substitution is legal as soon as $R_\alpha \subseteq [\mathsf{id}]$ and $R_\alpha(\mathbf{x}) = R_{\alpha\tau}(\mathbf{x})$ for all $\alpha \in \mathcal{S}(d)$, $\tau \in \mathcal{T}(d)$ and every $\mathbf{x} \in [n]^d$ such that $\mathbf{x}_\tau = \mathbf{x}$. Then, Lemma 7.6 gives the translation of R-atomic formulas into formulas expressed in term of the R_α's. All in all, we get the equivalence of the initial formula Φ to the following:

$$\exists (R_\alpha)_{\alpha \in \mathcal{S}(d)} \left\{ \begin{array}{l} \forall \mathbf{x} \bigwedge\limits_{\alpha \in \mathcal{S}(d)} (R_\alpha(\mathbf{x}) \to \mathbf{x} \in [\mathsf{id}]) \wedge \\ \forall \mathbf{x} \bigwedge\limits_{\alpha \in \mathcal{S}(d)} \bigwedge\limits_{\tau \in \mathcal{T}(d)} (\mathbf{x}_\tau = \mathbf{x} \to (R_\alpha(\mathbf{x}) \leftrightarrow R_{\alpha\tau}(\mathbf{x}))) \wedge \\ \forall \mathbf{x} \bigwedge\limits_{\alpha \in \mathcal{S}(d)} \big(\mathbf{x} \in [\alpha] \to \varphi_\alpha(\mathbf{x}, (R_\gamma)_{\gamma \in \mathcal{S}(d)})\big) \end{array} \right\} \tag{25}$$

where each φ_α is obtained from φ by the substitutions:

- $R(\mathbf{x}_\beta) \rightsquigarrow R_{\beta^{-1}\alpha}(x_\alpha)$
- $R(\mathbf{x}^{(i)}) \rightsquigarrow \left\{ \begin{array}{l} \left(x_i = x_{\alpha_d} \wedge R_{\alpha\tau_{\alpha^{-1}(i),d}}((x_\alpha)^{(d)})\right) \vee \\ \bigvee_{i \le k < d} \left(x_i = x_{\alpha_k} \neq x_{\alpha_{k+1}} \wedge R_{\alpha\tau_{\alpha^{-1}(i),k}}((x_\alpha)^{(k)})\right) \end{array} \right\}$

The first two conjuncts of (25) ensure that the family $(R_\alpha)_{\alpha\in\mathcal{S}(d)}$ encodes a relation R (see Lemma 7.5); the third interprets assertions of the form $R(\mathbf{x}_\beta)$ and $R(\mathbf{x}^{(i)})$ according to the modalities described in Lemma 7.6. Because of permutability of the conjunction and the universal quantifier, this third conjunct can be rewritten:

$$\bigwedge_{\alpha\in\mathcal{S}(d)} \forall \mathbf{x} : \mathbf{x} \in [\alpha] \to \varphi_\alpha(\mathbf{x}, (R_\gamma)_{\gamma\in\mathcal{S}(d)}) \tag{26}$$

For a fixed conjunct in (26), i.e. for a fixed α, all atomic subformulas of φ_α built on the R_γ's have the form $R_\gamma(\mathbf{x}_\alpha)$ or $R_\gamma((\mathbf{x}_\alpha)^{(i)})$ for some $\gamma \in \mathcal{S}(d)$ and some $i \in \{1,\dots,d\}$. Hence, the substitution of variables $\mathbf{x}/\mathbf{x}_\alpha$ allows to write such a conjunct as: $\forall \mathbf{x} : \mathbf{x} \in [\mathsf{id}] \to \tilde{\varphi}_\alpha$ where $\tilde{\varphi}_\alpha \equiv \varphi_\alpha(\mathbf{x}/\mathbf{x}_\alpha)$ only involves (R_γ)-subformulas of the form $R_\gamma(\mathbf{x})$ or $R_\gamma(\mathbf{x}^{(k)})$ for some $\gamma \in \mathcal{S}(d)$ and $k \in \{1,\dots,d\}$. Finally, the initial formula Φ is proved equivalent to the following formula Ψ, whose all R-atoms agrees with the sorted property (Condition (3*b*) of Definition 6.2):

$$\Psi \equiv \exists (R_\alpha)_{\alpha\in\mathcal{S}(d)} \left\{ \begin{array}{l} \forall \mathbf{x} \bigwedge_{\alpha\in\mathcal{S}(d)} (R_\alpha(\mathbf{x}) \to \mathbf{x} \in [\mathsf{id}]) \wedge \\ \forall \mathbf{x} \bigwedge_{\alpha\in\mathcal{S}(d)} \bigwedge_{\tau\in\mathcal{T}(d)} (\mathbf{x}_\tau = \mathbf{x} \to (R_\alpha(\mathbf{x}) \leftrightarrow R_{\alpha\tau}(\mathbf{x}))) \wedge \\ \forall \mathbf{x} \bigwedge_{\alpha\in\mathcal{S}(d)} \left(\mathbf{x} \in [\mathsf{id}] \to \tilde{\varphi}_\alpha(\mathbf{x}, (R_\gamma)_{\gamma\in\mathcal{S}(d)})\right) \end{array} \right\} \tag{27}$$

The proof is not yet completed: one could refer to Definition 6.4 (3) to verify that neither the relation $=$ nor the relation $<$ can be involved in an $\mathrm{ESO}(\forall^d, \text{arity } d, \text{half-sorted})$-formula. However, these two predicates still appear in formula (27) (the inequalities are hidden in the expression "$\mathbf{x} \in [\mathsf{id}]$" that abbreviates the formula $x_1 \le \dots \le x_d$), and it remains to get rid of them. To this end, we introduce two new binary relation symbols E and $P_<$ that will be forced to coincide with $=$ and $<$ by use of inductive constraints written with the successor function. But the assignment of maintaining the sorted property precludes us from writing formulas that we would be naturally minded to invoke when formalizing these constraints. For instance, the formulas $E(x_1, x_1)$ and $E(x_1, x_2) \leftrightarrow E(\mathrm{succ}(x_1), \mathrm{succ}(x_2)$ contain non sorted atoms and hence don't fit our syntactical restrictions. Therefore, we introduce two additionnal binary relation symbols, S and $P_>$, intended to simulate, respectively, the successor function and the strict linear order opposite to $<$. It appears that we can stipulate the interpretations of E and S by a simultaneous inductive scheme expressed in sorted fashion thanks to the formula below:

$$\Theta_1(E,S) \equiv \forall x_1, x_2 \left\{ \begin{array}{ll} \min(x_1) \to (\min(x_2) \leftrightarrow E(x_1,x_2)) & \wedge \\ \min(x_2) \to (\min(x_1) \leftrightarrow E(x_1,x_2)) & \wedge \\ \neg\max(x_2) \to (E(x_1,x_2) \leftrightarrow S(x_1, \mathrm{succ}(x_2)) & \wedge \\ S(x_1,x_2) \leftrightarrow E(\mathrm{succ}(x_1), x_2) & \end{array} \right\}.$$

In a similar way, $P_<$ and $P_>$ are compelled to fit their intended meaning *via* the following formula, in which we assume the interpretation of E has been previously fixed (by Θ_1):

$$\Theta_2(E,P_<,P_>) \equiv \forall x_1,x_2 \left\{ \begin{array}{rcll} (E(x_1,x_2)\vee P_<(x_1,x_2)) & \rightarrow & (\neg\max(x_2)\rightarrow P_<(x_1,\mathrm{succ}(x_2))) & \wedge \\ (E(x_1,x_2)\vee P_>(x_1,x_2)) & \rightarrow & (\neg\max(x_1)\rightarrow P_>(\mathrm{succ}(x_1),x_2)) & \wedge \\ P_<(x_1,x_2) & \rightarrow & (\neg P_>(x_1,x_2)\wedge\neg E(x_1,x_2)) & \wedge \\ P_>(x_1,x_2) & \rightarrow & \neg E(x_1,x_2) & \end{array} \right\}.$$

Clearly, the formula Ψ displayed in (27) can be rewritten in $\exists E,S,P_<,P_> : \tilde{\Psi}\wedge\Theta_1\wedge\Theta_2$, where $\tilde{\Psi}$ is obtained from Ψ by the substitutions:

$$x_i = x_j \rightsquigarrow E(x_i,x_j),\ x_i < x_j \rightsquigarrow P_<(x_i,x_j)\ \text{ and } x_j < x_i \rightsquigarrow P_>(x_i,x_j)$$

for any $i < j$. The resulting formula respects the sorting property. At last, notice that one can subtitute d-ary relation symbols for E, S, $P_<$ and $P_>$ without spoiling the sorted property, as we did it in the proof of Fact 7.1(b). □

All in all, the results obtained in Section 7 can be recapitulated as follows:

> On $(d-1)$-coordinate pictures, each ESO($\forall^d$, arity d)-formula can be written under the form:
>
> $$\Phi \equiv \exists\mathbf{R}\forall\mathbf{x} \bigwedge\bigvee \pm\left\{\ \min(x),\ \max(x),\ Q_s(\mathbf{x}_\iota),\ R(\mathbf{x}),\ R(\mathbf{x}^{(i)})\ \right\}. \qquad (28)$$
>
> Here, R is d-ary and belongs to $\mathbf{R}$, $s\in\Sigma$ and Q_s has arity $d-1$, $x\in\mathbf{x}$, $\iota\in\mathcal{I}(d-1,d)$, $i\in[d]$.

8. Sorting input relations

We now embark on the last step of our "localization" of ESO(var d)-formulas. Thanks to the results proved so far, it suffices to prove the inclusion, to be stated in Proposition 8.6: on coordinate structures of dimension $d-1$,

$$\mathrm{ESO}(\forall^d, \text{arity } d, \text{half-sorted}) \subseteq \mathrm{ESO}(\forall^d, \text{arity } d, \text{sorted}).$$

With definitions 6.4 and 6.2 in mind, our purpose can be very simply defined: given a formula Φ of the form described in Equation (28), we want to replace by a sorted atom each atom of the form $Q(\mathbf{x}_\iota)$, where $Q\in(Q_s)_{s\in\Sigma}$ and $\iota\in\mathcal{I}(d-1,d)$, $\iota\neq\mathsf{id}$. In the vein of Section 7, we aim at defining a tuple of relations $(Q_\iota)_{\iota\in\mathcal{I}(d-1,d)}$, in such a way that $Q_\iota(\mathbf{x}) = Q(\mathbf{x}_\iota)$ for each $\mathbf{x}$. Clearly, such relations will allow to write Φ under the desired form, by substituing each subformula $Q(\mathbf{x}_\iota)$ with $Q_\iota(\mathbf{x})$. Of course, the definition of the Q_ι's must be logically formulated in our syntactical restrictions, that is, without involving any $\mathbf{x}_\alpha$ such that $\alpha\neq\mathsf{id}$. Furthermore, the way we carried out this strategy in Section 7 is no more avalaible. Indeed, there we sorted atoms $R(\mathbf{x}_\alpha)$ build on some *existentially quantified* d-ary relations R by *suppressing* R in favour of some new existentially quantified relations R_α, $\alpha\in\mathcal{S}(d)$. (See proof of Proposition 7.7.) Of course, we can't operate like this with the relation Q, which is *part of the input*.

To give a hint of the method used in the present section, let's start with an example.

8.1. An easy case

Consider the case where $d = 2$. We deal with two first-order variables x and y and we only accept atoms of the form $Q(x)$, $R(x,y)$, $R(\mathrm{succ}(x),y)$ and $R(x,\mathrm{succ}(y))$ for any input unary relation Q and any guessed binary relation R. How can we tackle occurences of some atom $Q(y)$ in the formula ? A natural idea is to define a new binary relation Q_2 in such a way that $Q_2(x,y) = Q(y)$ holds for any x,y. (We denote it Q_2 to refer both to Q and to the projection of (x,y) on its *second* component.) Hence, we set:

$$Q_2 = \{(x,y) : Q(y)\}.$$

Thus, any atom $Q(y)$ could be replaced by the sorted atom $Q_2(x,y)$. But the logical definition of Q_2 with our syntactical constraints compels to introduce an additional binary relation T that will be used as a buffer to *transport* the information $Q(y)$ into the expression $Q_2(x,y)$. Informally, we can set

$$T = \{(x,y) : Q(x+y)\}.$$

Clearly, T is inductively defined from Q by the assertions $T(x,0) = Q(x)$ and $T(x+1,y) = T(x,y+1)$. Besides, Q_2 is defined from T by $Q_2(0,y) = T(0,y)$ and $Q_2(x,y) = Q_2(x+1,y)$. All these assertions can be rephrazed in our logical framework, with the following formulas:

$$\forall x,y \left\{ \begin{array}{rcl} \min(y) & \to & (T(x,y) \leftrightarrow Q(x)) \ \wedge \\ (\neg\max(x) \wedge \neg\max(y)) & \to & T(\mathrm{succ}(x),y) \leftrightarrow T(x,\mathrm{succ}(y)) \end{array} \right\};$$

$$\forall x,y \left\{ \begin{array}{l} \min(x) \to (Q_2(x,y) \leftrightarrow T(x,y)) \ \wedge \\ Q_2(x,y) \leftrightarrow Q_2(\mathrm{succ}(x),y) \end{array} \right\}.$$

It just remains to insert this defining formulas in the initial formula Φ to be normalized, and to replace each occurrence of $Q(y)$ by $Q_2(x,y)$. Of course, such a construction has to be carried on for each input unary relation Q.

8.2. General case: the sliding puzzle

When the dimension d exceeds 2, the construction is more intricate. For a given input relation Q, there are now numerous sources of corruption of the sorted property. For instance, if $d = 4$ then Q has arity 3 and we may enconter many kinds of non sorted Q-atoms in the formula to be normalized: $Q(x_2,x_3,x_1)$, $Q(x_1,x_4,x_2)$, $Q(x_2,x_3,x_4)$, etc. Dealing with these different atoms necessitate to introduce two new series of predicates, $(Q_\alpha)_{\alpha\in\mathcal{S}(d)}$ and $(T_\alpha)_{\alpha\in\mathcal{S}(d)}$, whose definitions are interconnected. In particular, the definition of the T_α's must be done step-by-step, according to an inductive process by which a new relation T_β is built from a yet-defined relation T_α, where α and β differ by exactly one transposition (i.e., $\beta = \alpha\tau$ for some $\tau \in \mathcal{T}(d)$). The key point of the proof will be to organize a traversal of the set of permutations which allows this recursive procedure of definition. In particular, it will be necessary that two consecutive transpositions labeling this traversal share a common position – that is, they should be written $(i\,j)$ and (jk) for some $i,j,k \in [d]$ –, in the same way a move from one position to another in a *sliding puzzle* always involves the place left vacant by the previous move.

Given $i,j \in \{1,\dots,d\}$ and $\alpha \in \mathcal{S}(d)$, we denote by $(i\,j)$ the transposition that exchanges i and j,[4] and by $\alpha(i\,j)$ the composition of α and $(i\,j)$. It is well-known that each permutation α can be written as a product of transpositions,

$$\alpha = (u_1v_1)(u_2v_2)\dots(u_pv_p). \tag{29}$$

[4] rather than $\tau_{i,j}$, as in the previous section.

Besides, for any transposition (uv) and any r not in $\{u,v\}$, it holds $(uv) = (ru)(uv)(vr)$. This yields two consequences of interest for the decomposition (29):

- First, we can assume that $u_1 = d$ (or equivalently, $v_1 = d$), since if $d \notin \{u_1, v_1\}$, then we can write $\alpha = (du_1)(u_1v_1)(v_1d)(u_2v_2)\dots(u_pv_p)$.
- Moreover, (29) can be further refined, by demanding that $v_i = u_{i+1}$ for each i. For if some sequence $(ab)(cd)$ with $b \notin \{c,d\}$ occurs in (29), it can be replaced by $(ab)(bc)(cd)(db)$, and a left-to-right iteration of such rewritings along the factorization yields the desired property.

Finally, does mean suppressing useless sequences as $(ab)(ba)$, each $\alpha \in \mathcal{S}(d)$ can be written

$$\alpha = (du_1)(u_1u_2)(u_2u_3)\dots(u_{k-2}u_{k-1})(u_{k-1}u_k), \tag{30}$$

where for each i, the terms u_i, u_{i+1} and u_{i+2} are pairwise distinct elements of $\{1,\dots,d\}$. We call *alternated factorization of α* such a decomposition.

A permutation α admits several alternated factorizations, and we want to single out one of them for each α, in order to allow an inductive reasoning built on this particular decomposition. There is no canonical way to perform this task. In the following lemma, we roughly describe one arbitrary choice. It refers to the Cayley graph $\mathcal{G}_d$ of the symmetric group $\mathcal{S}(d)$ with respect to the set of generators $\mathcal{T}(d)$. Recall that $\mathcal{G}_d$ is the graph of domain $\mathcal{S}(d)$ whose edges are all pairs of permutations $(\alpha, \alpha(ij))$, for $i \neq j$ in $\{1,\dots,d\}$.

Lemma 8.1. *There exists an oriented tree $\mathcal{T}_d$ covering $\mathcal{G}_d$ that is rooted at* id *and such that each $\mathcal{T}_d$-path starting at* id*, say $\mathsf{id}\alpha_1\dots\alpha_p$, corresponds to an alternated factorization of α_p.*

Proof. Trees $\mathcal{T}_d$ for $d \geq 2$ are defined inductively. For $d = 2$, there is a unique such tree: $(12) \to (21) = (12)(21)$. So, assume we are given $\mathcal{T}_{d-1}$ and carry on the construction of $\mathcal{T}_d$ as follows:

(a) First, view each permutation $\alpha = \alpha_1\dots\alpha_{d-1} \in \mathcal{S}(d-1)$ as a permutation of $\{2,\dots,d\}$ by renaming α_i as $\alpha_i + 1$. That is, replace $\alpha = \alpha_1\dots\alpha_{d-1}$ by $\alpha^+ = (\alpha_1+1)\dots(\alpha_{d-1}+1)$.

(b) Then, replace each such α^+ by $\beta = \beta_1\dots\beta_d \in \mathcal{S}(d)$ defined by: $\beta_1 = 1$ and $\beta_i = [\alpha^+]_{i-1} = \alpha_{i-1}+1$ for $i > 1$. Thus, $\mathcal{T}_{d-1}$ now covers the set of permutations $\beta \in \mathcal{S}(d)$ that fulfill $[\beta]_1 = 1$.

(c) For each node β thus obtained, create a new node labelled by the composition of β with the transposition $(1d)$ – that is by the permutation $\beta(1d)$ – and create an edge $\beta \to \beta(1d)$.

(d) Finally, link each such node $\beta(1d)$ to $d-2$ new nodes $\beta(1d)(di)$, for $i = 2,\dots,d-1$.

In Fig 2, we display the steps of the construction of $\mathcal{T}_4$ from $\mathcal{T}_2$. Letters (a), …, (d) in the figure refer to the above items. The correction of the method on this example is clear. We leave it to the reader to verify that it generalises to any d. □

This lemma allows us to choose, for each $\alpha \in \mathcal{S}(d)$, one specified alternated factorization of α: it is the decomposition $(di_1)(i_1i_2)\dots(i_{k-1}i_k)$ corresponding to the unique path from id to α in $\mathcal{T}_d$. We denote by $\mathsf{id}.di_1.i_1i_2.\dots.i_{k-1}i_k$ this particular factorization. And when this path until α can be continued *in* $\mathcal{T}_d$ to some permutation $\alpha(i_ki_{k+1})$, we denote by $\alpha.i_ki_{k+1}$ this last permutation. For instance, in the example displayed in Fig 2, we can write $2143 = 4123.13$ and $3124 = 4123.14$ while $4321 = 4123(24)$ *cannot be written* 4123.24. Notice furthermore that the integers i_k and i_{k+1} are ordered in the notation $\alpha.i_ki_{k+1}$ (unlike in the notation $\alpha(i_ki_{k+1})$): we place in first position the integer i_k involved in the last transposition leading to α (with $i_k = d$ if $\alpha = \mathsf{id}$). All in all, the reader is invited to keep in mind the numerous presuppositions attached to the notation $\alpha.uv$: the statement $\beta = \alpha.uv$ means:

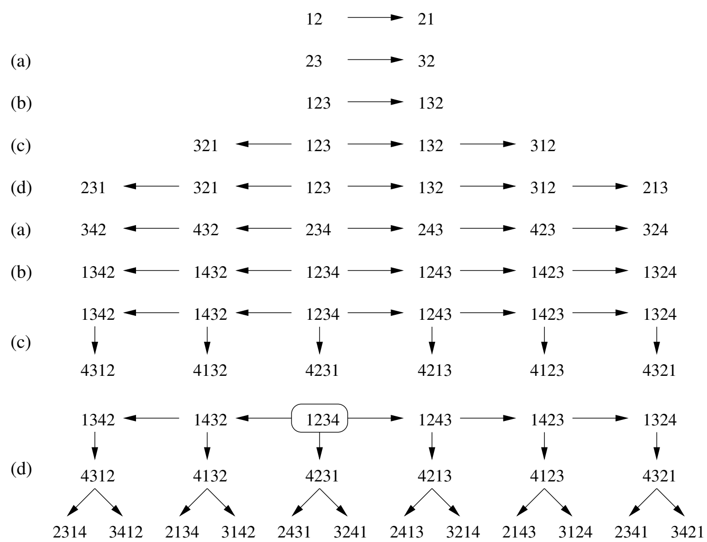

Figure 2: Construction of $\mathcal{T}_4$ from $\mathcal{T}_2$. The result is an oriented tree rooted at id, spanning $\mathcal{S}(4)$, whose all pathes from id are alternated.

- $\beta = \alpha(uv)$;
- (α,β) is an edge of $\mathcal{T}_d$;
- either $\alpha = \mathsf{id}$ and $u = d$, or $\alpha = \gamma.tu$ for some $\gamma \in \mathcal{S}(d)$ and some $t \neq u$ in $\{1,\dots,d\}$.

Let us now come to a straightforward remark connecting d-tuples to alternated factorizations. First, remember that for $\mathbf{x} = (x_1,\dots,x_d)$, $i \in \{1,\dots,d\}$ and $\alpha,\beta \in \mathcal{S}(d)$,

- we denoted by $[\mathbf{x}]_i$ the i^{th} component of $\mathbf{x}$;
- we defined $\mathbf{x}_\alpha$ as the d-tuple $(x_{\alpha_1},\dots,x_{\alpha_d})$;
- we noticed that $(\mathbf{x}_\alpha)_\beta = \mathbf{x}_{\alpha\beta}$.

(See Definition 6.3.)

Fact 8.2. *For any $\mathbf{x} \in [n]^d$, $\alpha \in \mathcal{S}(d)$, and $i,j \in \{1,\dots,d\}$:*

(a) $\mathbf{x}_{\alpha.ij} = (\mathbf{x}_\alpha)_{(ij)}$.

(b) $[\mathbf{x}_{\alpha.ij}]_j = [\mathbf{x}]_d$.

Proof. (*a*) From $\alpha.ij = \alpha(ij)$ and $\mathbf{x}_{\alpha\beta} = (\mathbf{x}_\alpha)_\beta$ (see Definition 6.3(*iv*)). (*b*) By induction on α: if $\alpha = \mathsf{id}$, then necessarily $i = d$ and $[\mathbf{x}_{\mathsf{id}.dj}]_j = [\mathbf{x}]_d$ clearly holds. Otherwise, $[\mathbf{x}_{\alpha.ij}]_j = [(\mathbf{x}_\alpha)_{ij}]_j = [\mathbf{x}_\alpha]_i = [\mathbf{x}]_d$, by induction hypothesis. □

Given a d-tuple $\mathbf{x} = (x_1, \ldots, x_d)$ of first-order variables, we denote by $\mathbf{x}^-$ the $(d-1)$-tuple obtained from $\mathbf{x}$ by erasing its last component. That is,

$$(x_1, \ldots, x_{d-1}, x_d)^- := (x_1, \ldots, x_{d-1}).$$

In particular, for $\alpha \in \mathcal{S}(d)$, we denote by $\mathbf{x}_\alpha^-$ the $(d-1)$-tuple $(x_{\alpha_1}, \ldots, x_{\alpha_{d-1}})$. Each $(d-1)$-tuple build upon the d variables $x_1, \ldots, x_d$ can clearly be written $\mathbf{x}_\alpha^-$ for some $\alpha \in \mathcal{S}(d)$. Therefore, the occurence of a non-sorted atom in Φ has the form $Q(\mathbf{x}_\alpha^-)$ for some $Q \in (Q_s)_{s\in\Sigma}$ and some $\alpha \in \mathcal{S}(d)$, and the purpose of this section amounts to rewrite each such occurence $Q(\mathbf{x}_\alpha^-)$ as $Q'(\mathbf{x})$ for some well chosen relation Q'.

Definition 8.3. *Given a $(d-1)$-ary relation Q and two family of d-ary relations, $(T_\alpha)_{\alpha\in\mathcal{S}(d)}$ and $(Q_\alpha)_{\alpha\in\mathcal{S}(d)}$, we say that $(T_\alpha, Q_\alpha)_{\alpha\in\mathcal{S}(d)}$ is a d-simulation of Q if the following axioms hold, for any $\alpha \in \mathcal{S}(d)$ and any $i, j \leq d$ such that $\alpha.ij$ is defined, and for any d-tuple $\mathbf{x}$ of variables:*

(A_1) $T_{\mathsf{id}}(\mathbf{x}) = Q(\mathbf{x}^-)$ *if* $[\mathbf{x}]_d = 0$.

(A_2) $T_{\alpha.ij}(\mathbf{x}) = T_{\alpha.ij}(\mathbf{x}_{(ij)})$.

(A_3) $T_\alpha(\mathbf{x}) = T_{\alpha.ij}(\mathbf{x})$ *if* $[\mathbf{x}]_i = 0$.

(A_4) $Q_{\mathsf{id}}(\mathbf{x}) = Q(\mathbf{x}^-)$.

(A_5) $Q_{\alpha.ij}(\mathbf{x}) = T_{\alpha.ij}(\mathbf{x})$ *if* $[\mathbf{x}]_j = 0$.

(A_6) $Q_{\alpha.ij}(\mathbf{x})$ *doesn't depend on* $[\mathbf{x}]_j$.

Lemma 8.4. *Let $(T_\alpha, Q_\alpha)_{\alpha\in\mathcal{S}(d)}$ be a d-simulation of some $(d-1)$-ary predicate Q. For any $\mathbf{x} \in [n]^d$ and $\alpha \in \mathcal{S}(d)$: $Q_\alpha(\mathbf{x}_\alpha) = Q(\mathbf{x}^-)$.*

Proof. Let us first prove that for any $\mathbf{x} \in [n]^d$ and $\alpha \in \mathcal{S}(d)$:

$$[\mathbf{x}]_d = 0 \Rightarrow T_\alpha(\mathbf{x}_\alpha) = Q(\mathbf{x}^-). \tag{31}$$

We proceed by induction on α. If $\alpha = \mathsf{id}$, (31) follows from (A_1). Given a non-identique permutation $\alpha.ij$, we have:

$$\begin{aligned} T_{\alpha.ij}(\mathbf{x}_{\alpha.ij}) &= T_{\alpha.ij}((\mathbf{x}_\alpha)_{(ij)}) \text{ by Fact 8.2-(a).} \\ &= T_{\alpha.ij}(\mathbf{x}_\alpha) \text{ by } (\mathsf{A}_2). \end{aligned}$$

But $[\mathbf{x}_\alpha]_i = [(\mathbf{x}_\alpha)_{(ij)}]_j = [\mathbf{x}_{\alpha.ij}]_j$ and hence, by Fact 8.2-(b): $[\mathbf{x}_\alpha]_i = [\mathbf{x}]_d = 0$. Therefore we can resume the above sequence of equalities with:

$$\begin{aligned} T_{\alpha.ij}(\mathbf{x}_{\alpha.ij}) &= T_\alpha(\mathbf{x}_\alpha) \text{ by } (\mathsf{A}_3) \text{ since } [\mathbf{x}_\alpha]_i = 0. \\ &= Q(\mathbf{x}^-) \text{ by induction hypothesis.} \end{aligned}$$

This completes the proof of (31)

Let us now prove the equality $Q_\alpha(\mathbf{x}_\alpha) = Q(\mathbf{x}^-)$. If $\alpha = \mathsf{id}$, the result comes from (A_4). For a non-identique permutation $\alpha.ij$, we have to prove $Q_{\alpha.ij}(\mathbf{x}_{\alpha.ij}) = Q(\mathbf{x}^-)$ for any tuple $\mathbf{x} \in [n]^d$. First notice that we can restrict without loss of generality to the case where $[\mathbf{x}]_d = 0$. Indeed, we can otherwise consider the tuple $\mathbf{y}$ obtained from $\mathbf{x}$ by setting $[\mathbf{x}]_d$ to 0. (That is, $\mathbf{y}$ only differs from $\mathbf{x}$ by its d^{th} component, which is null.) Clearly, $\mathbf{y}^- = \mathbf{x}^-$. Besides, the j^{th} component of $\mathbf{x}_{\alpha.ij}$ is $[\mathbf{x}]_d$, from Fact 8.2-(b). Similarly, the j^{th} component of $\mathbf{y}_{\alpha.ij}$

is $[\mathbf{y}]_d$. Hence, the tuples $\mathbf{x}_{\alpha.ij}$ and $\mathbf{y}_{\alpha.ij}$ coincide on each component of rank distinct from j. Therefore $Q_{\alpha.ij}(\mathbf{x}_{\alpha.ij}) = Q_{\alpha.ij}(\mathbf{y}_{\alpha.ij})$ by (A_6) and we get: $Q_{\alpha.ij}(\mathbf{x}_{\alpha.ij}) = Q(\mathbf{x}^-)$ iff $Q_{\alpha.ij}(\mathbf{y}_{\alpha.ij}) = Q(\mathbf{y}^-)$. Thus, we can assume $[\mathbf{x}]_d = 0$. It follows $[\mathbf{x}_{\alpha.ij}]_j = [\mathbf{x}]_d = 0$, by Fact 8.2-(b), and hence:

$$\begin{array}{rcl} Q_{\alpha.ij}(\mathbf{x}_{\alpha.ij}) & = & T_{\alpha.ij}(\mathbf{x}_{\alpha.ij}) \text{ by } (\mathsf{A}_5), \text{ since } [\mathbf{x}_{\alpha.ij}]_j = 0. \\ & = & Q(\mathbf{x}^-) \text{ by } (31), \text{ since } [\mathbf{x}]_d = 0. \end{array}$$

The proof is complete. □

Lemma 8.5. *Let Q be a $(d-1)$-ary relation and $(T_\alpha)_{\alpha\in\mathcal{S}(d)}$, $(Q_\alpha)_{\alpha\in\mathcal{S}(d)}$ be two tuple of d-ary relations satisfying, for each d-tuple $\mathbf{x} = (x_1, \ldots, x_d)$:*

(F_1) $min(x_d) \to (T_{id}(\mathbf{x}) \leftrightarrow Q(\mathbf{x}^-))$.

(F_2) $\big(\neg max(x_i) \wedge \neg max(x_j)\big) \to \big(T_{\alpha.ij}(\mathbf{x}^{(i)}) \leftrightarrow T_{\alpha.ij}(\mathbf{x}^{(j)})\big)$.

(F_3) $min(x_i) \to \big(T_\alpha(\mathbf{x}) \leftrightarrow T_{\alpha.ij}(\mathbf{x})\big)$.

(F_4) $Q_{id}(\mathbf{x}) \leftrightarrow Q(\mathbf{x}^-)$.

(F_5) $min(x_j) \to \big(Q_{\alpha.ij}(\mathbf{x}) \leftrightarrow T_{\alpha.ij}(\mathbf{x})\big)$.

(F_6) $Q_{\alpha.ij}(\mathbf{x}) \leftrightarrow Q_{\alpha.ij}(\mathbf{x}^{(j)})$.

Then $(T_\alpha, Q_\alpha)_{\alpha\in\mathcal{S}(d)}$ is a d-simulation of Q. Furthermore, each Q admits such a d-simulation fulfilling (F_1)…(F_6)

Proof. Clearly, the formula (F_i) is a mere transcription of the axiom (A_i) for each $i \neq 2$. We have to prove that (F_2) implies (A_2). Formula (F_2) yields that $T_{\alpha.ij}(\mathbf{x})$ has the same value on tuples of the form

$$\mathbf{x} = (\mathbf{u}, x+c, \mathbf{v}, y-c, \mathbf{w}),$$

for any $c \leq \min\{n-1-x, y\}$, where $x+c$ (resp. $y-c$) is the component of rank i (resp. j) of $\mathbf{x}$. (That is: $\mathbf{u} \in [n]^{i-1}$, $\mathbf{v} \in [n]^{j-i-1}$ and $\mathbf{w} \in [n]^{d-j}$.) In other words, the value of $T_{\alpha.ij}(\mathbf{x})$ depends on $[\mathbf{x}]_i + [\mathbf{x}]_j$ rather than on the precise values of these two components. As a consequence, for any $\mathbf{u} \in [n]^{i-1}$, $\mathbf{v} \in [n]^{j-i-1}$, $\mathbf{w} \in [n]^{d-j}$ and any $x, y \in [n]$:

$$T_{\alpha.ij}(\mathbf{u}, x, \mathbf{v}, y, \mathbf{w}) = T_{\alpha.ij}(\mathbf{u}, y, \mathbf{v}, x, \mathbf{w}).$$

This is Axiom (A_2).

It remains to prove that such relations $(T_\alpha)_{\alpha\in\mathcal{S}(d)}$ and $(Q_\alpha)_{\alpha\in\mathcal{S}(d)}$ exist for every Q. To see this, assume we are given a $(d-1)$-ary relation Q and define the Q_α's and the T_α's as follows:

- $Q_\alpha(\mathbf{x}) = Q(\mathbf{x}^-_{\alpha^{-1}})$ for any $\mathbf{x} \in [n]^d$;
- $T_{\mathsf{id}}(\mathbf{x}) = Q_{id}(\mathbf{x})$ for any $\mathbf{x} \in [n]^d$;
- $T_{\alpha.ij}(\mathbf{u}, x, \mathbf{v}, y, \mathbf{w}) = Q_{\alpha.ij}(\mathbf{u}, x+y, \mathbf{v}, 0, \mathbf{w})$ for any $x, y \in [n]$, $\mathbf{u} \in [n]^{i-1}$, $\mathbf{v} \in [n]^{j-i-1}$ and $\mathbf{w} \in [n]^{d-j}$.

We leave it to the reader to check that the sequence $(T_\alpha, Q_\alpha)_{\alpha\in\mathcal{S}(d)}$ satisfies the formulas (F_1), …, (F_6) (and hence, is a d-simulation of Q). □

Proposition 8.6. *For any $d > 1$, $\mathrm{ESO}(\forall^d, \text{arity } d, \text{half-sorted}) \subseteq \mathrm{ESO}(\forall^d, \text{arity } d, \text{sorted})$ on coordinate structures of dimension $d-1$.*

Theorem 5.1, Proposition 7.7 and Proposition 8.6 can now be collected in the following result:

Theorem 8.7. *For any $d > 1$, $\mathrm{ESO}(\mathrm{var}\, d) = \mathrm{ESO}(\forall^d, \text{arity } d, \text{sorted})$ on coordinate structures of dimension $d-1$.*

All in all, our normalization process can be summarized as follows:

On $(d-1)$-coordinate structures, each $\mathrm{ESO}(\mathrm{var}\, d)$-formula can be written under the form:

$$\Phi \equiv \exists \mathbf{R} \forall \mathbf{x} \bigwedge \bigvee \pm \left\{ \ \min(x_i),\ \max(x_i),\ Q(\mathbf{x}^-),\ R(\mathbf{x}),\ R(\mathbf{x}^{(i)}) \ \right\} \tag{32}$$

Here, $\mathbf{R}$ (resp. $\mathbf{Q}$) is a list of relation symbols of arity d (resp. $d-1$), $\mathbf{x} = (x_1, \ldots, x_d)$, $i \in [d]$, $Q \in (Q_s)_{s \in \Sigma}$, $R \in \mathbf{R}$.

Remark 8.8.

1. *The proof of Proposition 8.6 crucially involves the possibility to use one of the d dimensions that is left unoccupied by the $(d-1)$-ary input relation symbols. This free dimension acts as a buffer that enabled us to play the above mentioned* sliding puzzle*. Therefore, the result is* a fortiori *valid when the input relations have arity less than $d-1$. In other terms, the statement of Proposition 8.6, and hence that of Theorem 8.7 , actually holds on coordinate structures of dimension $k < d$.*
2. *Besides, one can assume that the successor function occuring in formulas displayed in (32) only applies to arguments that are not maximal, or alternatively, that $\mathrm{succ}(n) = n$, which means that the interpretation of the successor function symbol is the* noncyclic *successor* instead of *the cyclic one.*

9. Recapitulation of the results: power/limits of coordinate/pixel encodings

9.1. *Characterization of* NLIN_{ca}

After all the previous normalizations of logics over the coordinate encoding of pictures, we are now in a position to prove the expected characterization of $\mathrm{NLIN}_{\mathrm{ca}}$.

Theorem 9.1. *For any $d > 0$ and any d-language L,*

$$L \in \mathrm{NLIN}^d_{ca} \Leftrightarrow \mathsf{coord}^d(L) \in \mathrm{ESO}(\mathrm{var}\, d+1).$$

This theorem will straightforwardly proceed from Proposition 4.10, Theorem 8.7, and from Proposition 9.2 below:

Proposition 9.2. *For any $d > 0$ and any d-language L,*

$$\mathsf{coord}^d(L) \in \mathrm{ESO}(\mathrm{var}\, d+1) \Rightarrow L \in \mathrm{NLIN}^d_{ca}.$$

Proof. Let Φ be the formula characterizing $\mathsf{coord}^d(L)$. From Theorem 8.7 (see also Equation (32)), we can assume that Φ has the form:

$$\Phi \equiv \exists \mathbf{R} \forall \mathbf{x} \forall t \bigwedge \bigvee \pm \left\{ \begin{array}{l} \min(x_i),\ \max(x_i),\ \min(t),\ \max(t), \\ Q(\mathbf{x}),\ R(\mathbf{x}, t),\ R(\mathbf{x}^{(i)}, t),\ R(\mathbf{x}, \mathrm{succ}(t)) \end{array} \right\}.$$

for $i \in [d]$, $Q \in (Q_s)_{s\in\Sigma}$ and $R \in \mathbf{R}$. Moreover, according to Remark 8.8 we can assume without loss of generality that the succ symbol is interpreted as the *noncyclic* successor function.

The key point is that sentence Φ can be checked in $O(n)$ steps (for an input picture of domain $[n]^d$) by a *local and parallel nondeterministic* process. More precisely, it is easy to construct a d-automaton which uses the following informal but intuitive algorithm to check whether $\mathsf{coord}^d(p) \models \Phi$, for any input picture $p : [n]^d \to \Sigma$:

For $t = 1, 2, \ldots, n$, sequentially,
For all $\mathbf{a} \in [n]^d$, in parallel,

- *Guess (nondeterministically) the truth values of the atoms $R(\mathbf{a}, t)$;*
- *Check (deterministically) whether the values of the $R(\mathbf{a}, t)$'s are compatible*[5]
  - *with each other,*
  - *with the values of the input atoms $Q(\mathbf{a})$ and of the atoms $min(a_i)$ and $max(a_i)$,*
  - *with the "previous" values of atoms $R(\mathbf{a}, t-1)$ (provided $t > 1$);*
- *If some answer is "no" then Reject;*

Accept.

The process is correct because at each moment the cellular automaton at cell $\mathbf{a} \in [n]^d$ has only to consider the fixed number of information bits that the point $\mathbf{a} = (a_1, \ldots, a_d)$ and its d *neighbor* points $\mathbf{a}^{(i)} = (a_1, \ldots, a_{i-1}, \mathrm{succ}(a_i), a_{i+1}, \ldots, a_d)$ hold. Each of the n iterations of the first loop ("For $t = 1, 2, \ldots, n$") is performed in constant time, hence the total time is $O(n)$. □

9.2. Why do we need two encodings of pictures ?

In Section 3, we have established natural characterizations of REC^d *for pixel encoding* of pictures. The corresponding statements (Theorem 3.6 and Corollary 3.16) can be summed up as follows: for any $d > 0$ and any d-language L,

$$\begin{aligned} L \in \mathrm{REC}^d \quad & \Leftrightarrow \mathsf{pixel}^d(L) \in \mathrm{ESO}(\forall^1, \text{arity } 1) \\ & \Leftrightarrow \mathsf{pixel}^d(L) \in \mathrm{ESO}(\text{arity } 1) \\ & \Leftrightarrow \mathsf{pixel}^d(L) \in \mathrm{ESO}(\text{var } 1). \end{aligned} \tag{33}$$

Then, we have stated characterizations of $\mathrm{NLIN}^d_{\mathrm{ca}}$ *for coordinate encoding* of pictures (Theorem 9.1 and Theorem 8.7):

$$\begin{aligned} L \in \mathrm{NLIN}^d_{\mathrm{ca}} \quad & \Leftrightarrow \mathsf{coord}^d(L) \in \mathrm{ESO}(\forall^{d+1}, \text{arity } d+1, \text{sorted}) \\ & \Leftrightarrow \mathsf{coord}^d(L) \in \mathrm{ESO}(\forall^{d+1}, \text{arity } d+1) \\ & \Leftrightarrow \mathsf{coord}^d(L) \in \mathrm{ESO}(\text{var } d+1). \end{aligned} \tag{34}$$

One could legitimately ask whether a same encoding of pictures would permit characterizations of both REC^d and $\mathrm{NLIN}^d_{\mathrm{ca}}$. We now answer this question in its two aspects.

[5]That means: verify that the constraints of $\psi(\mathbf{a}, t)$ and $\psi(\mathbf{a}, t-1)$ (when $t > 1$) are satisfied together.

9.2.1. Pixel representation does not fit NLIN^d_{ca}

One can prove the following sequence of inclusions for pixel representation of d-picture languages with $d > 1$:

$$\mathrm{ESO}(\mathrm{var}\,1) = \mathrm{REC}^d \subsetneq \mathrm{NLIN}^d_{\mathrm{ca}} \subsetneq \mathrm{NTIME}^d_{ca}(n^d) \subseteq \mathrm{ESO}(\mathrm{var}\,2),$$

where $\mathrm{NTIME}^d_{ca}(n^d)$ denotes the class of d-picture languages recognized by nondeterministic d-dimensional cellular automata in time $O(n^d)$. Let us justify the (strict) inclusions that have not been proved beforehand in this paper.

- The strict inclusion $\mathrm{NLIN}^d_{\mathrm{ca}} \subsetneq \mathrm{NTIME}^d_{ca}(n^d)$ is a particular case of the nondeterministic time hierarchy theorem [5, 65] adapted to nondeterministic d-dimensional cellular automata, see [35] that has proved it for $d = 1$: we are convinced it can be generalized to any $d > 1$.
- The implication $L \in \mathrm{NTIME}^d_{ca}(n^d) \Rightarrow \mathsf{pixel}^d(L) \in \mathrm{ESO}(\mathrm{var}\,2)$ can be proved almost exactly as the particular implication $L \in \mathrm{NLIN}^1_{\mathrm{ca}} \Rightarrow \mathsf{pixel}^d(L) \in \mathrm{ESO}(\mathrm{var}\,2)$ among the recapitulated equivalences (34) above for word languages (adapt the proof of Proposition 4.10): consider the fact that a d-picture of side n contains n^d pixels and use the implicit lexicographic successor function of the set of pixels induced by the d successor functions, exactly as we did in Section 3.3.

Hence, the strict inclusions $\mathrm{ESO}(\mathrm{var}\,1) \subsetneq \mathrm{NLIN}^d_{\mathrm{ca}} \subsetneq \mathrm{ESO}(\mathrm{var}\,2)$ so established justify that no logic of the form $\mathrm{ESO}(\mathrm{var}\,k)$ – or, equivalently, $\mathrm{ESO}(\forall^k, \mathrm{arity}\,k)$) –, for any k, can characterize the class $\mathrm{NLIN}^d_{\mathrm{ca}}$ for pixel representation.

9.2.2. Coordinate representation does not fit recognizable picture languages

One can easily translate the logical characterization of REC^d in $\mathrm{ESO}(\forall^1, \mathrm{arity}\,1)$ for the pixel representation, into a characterization in $\mathrm{ESO}(\forall^d, \mathrm{arity}\,d, \mathrm{sorted}\,)$ for the coordinate representation. This is due to the following fact:

Fact 9.3. *$\mathsf{pixel}^d(L) \in \mathrm{ESO}(\forall^1, \mathrm{arity}\,1) \Leftrightarrow \mathsf{coord}^d(L) \in \mathrm{ESO}(\forall^d, \mathrm{arity}\,d, \mathrm{sorted}\,)$.*

Proof. To justify this equivalence, let us define a notion of *sorted* $\mathrm{ESO}(\forall^1, \mathrm{arity}\,1)$-formula, even if it is much less relevant that the corresponding definition for $d \geq 2$: An $\mathrm{ESO}(\forall^1, \mathrm{arity}\,1)$-formula Φ belongs to $\mathrm{ESO}(\forall^1, \mathrm{arity}\,1, \mathrm{sorted}\,)$ if it can be written $\Phi \equiv \exists \mathbf{U} \forall x \varphi$, where φ is a quantifier free formula of the form:

$$\varphi \equiv \bigwedge \bigvee \pm \Big\{\ Q(x),\ U(x),\ U(\mathrm{succ}_i(x)),\ \min_i(x),\ \max_i(x)\ \Big\} \tag{35}$$

Here, $\mathbf{U}$ is a list of unary relation symbols, $U \in \mathbf{U}$, Q belongs to the sequence $(Q_s)_{s\in\Sigma}$ of input unary relations, x is the unique first-order variable and $i \in [d]$.

Because of the proximity of pixel encoding and coordinate encoding of a d-picture $p : [n]^d \to \Sigma$, it is easy to associate with each $\Phi \equiv \exists \mathbf{U} \forall x \varphi$ of the form displayed in (35), an $\mathrm{ESO}(\forall^d, \mathrm{arity}\,d, \mathrm{sorted}\,)$-formula Ψ on coordinate signature, in such a way that $\mathsf{pixel}^d(p) \models \Phi \Leftrightarrow \mathsf{coord}^d(p) \models \Psi$. It suffices to set $\Psi = \exists \mathbf{R} \forall \mathbf{x} \psi(\mathbf{x})$ where:

- $\mathbf{R}$ is a list of d-ary relation symbols $(R_U)_{U\in\mathbf{U}}$, one-one associated with the unary relation variables $U \in \mathbf{U}$;
- $\psi(\mathbf{x})$ uses the list of d first-order variables $\mathbf{x} = (x_1, \ldots, x_d)$ and is obtained from $\varphi(x)$ by the substitutions:

– $Q(x) \rightsquigarrow Q(\mathbf{x})$, for $Q \in (Q_s)_{s\in\Sigma}$;
– $U(x) \rightsquigarrow R_U(\mathbf{x})$ and $U(\mathrm{succ}_i(x)) \rightsquigarrow R_U(\mathbf{x}^{(i)})$ for $U \in \mathbf{U}$ and $i \in [d]$;
– $\min_i(x) \rightsquigarrow \min(x_i)$ and $\max_i(x) \rightsquigarrow \max(x_i)$ for $i \in [d]$.

This definition clearly gives rise to a formula Ψ that belongs to $\mathrm{ESO}(\forall^d, \text{arity } d, \text{sorted})$. Furthermore, it immediatly yields the expected equivalence $\mathsf{pixel}^d(p) \models \Phi \Leftrightarrow \mathsf{coord}^d(p) \models \Psi$ for each d-picture p on Σ. To conclude the proof, it remains to verify that each $\mathrm{ESO}(\forall^1, \text{arity } 1)$-formula can be written in $\mathrm{ESO}(\forall^1, \text{arity } 1, \text{sorted})$, on pixel structures. We leave it to the reader. ◻

Mixing Fact 9.3 with the first equivalence in (33), we immediatly get:

Fact 9.4. $L \in \mathrm{REC}^d \Leftrightarrow \mathsf{coord}^d(L) \in \mathrm{ESO}(\forall^d, \text{arity } d, \text{sorted})$.

But requiring the sentence to be sorted seems rather artificial. In the characterization of $\mathrm{NLIN}_{\mathrm{ca}}$ (Theorem 9.1), we admitted this constraint as a necessary step of the proof, but it was crucial, in our point of view, that Theorem 8.7 provides us with a mean to circumvent it. What about the present case ? Can the sorted constraint be dropped in Fact 9.4 ? The forthcoming Proposition 9.7 will show that it cannot. In order to prove it, we first state a simple lemma about the so-called Mirror^d language.

Definition 9.5. *For $d \geq 2$, we denote by* Mirror^d *the set of pictures $p : [n]^d \to \Sigma$ that fulfill $p(\mathbf{a}) = p(\mathbf{a}_{(12)})$ for all $\mathbf{a} \in [n]^d$.*

(Remember that (12) is the transposition that swaps 1 an 2. Therefore, $\mathbf{a}_{(12)}$ is the tuple $\mathbf{a}$ in which the two first components have been interchanged – see Definition 6.3(*iv*). *E.g.*, $(5,3,4,6)_{(12)} = (3,5,4,6)$.)

Lemma 9.6. *For each $d \geq 2$, $\mathsf{coord}^d(\mathrm{Mirror}^d)$ belongs to $\mathrm{ESO}(\mathrm{var}\, d)$. However, $\mathrm{Mirror}^d \notin \mathrm{REC}^d$.*

Proof. Clearly, the set of structures $\mathsf{coord}^d(\mathrm{Mirror}^d)$ is defined by the following first-order sentence with d variables, hence it is in $\mathrm{ESO}(\mathrm{var}\, d)$:

$$\forall \mathbf{x} \bigwedge_{s\in\Sigma} (Q_s(\mathbf{x}) \leftrightarrow Q_s(\mathbf{x}_{(12)})).$$

The statement "$\mathrm{Mirror}^d \notin \mathrm{REC}^d$" can be proved by a reasoning very similar to that of Theorem 2.6 in Giammarresi et al. [23]. For sake of completeness, and because we will prove another adaptation of that result in Lemma 10.5, we now recall and adapt the proof explicitly. It essentially amounts to a counting argument. Let us sketch it:

Assume that the language Mirror^d is recognizable. Hence, it is the projection of a local language and can be written $\mathrm{Mirror}^d = \pi(\mathrm{Loc}) = \{\pi(p) : p \in \mathrm{Loc}\}$, for some local language $\mathrm{Loc} = L(\Delta_1, \ldots, \Delta_d)$ over an alphabet Γ, and some surjection $\pi : \Gamma \to \Sigma$ (see Definition 3.4). To get a contradiction, the trick is to mix two distinct pictures p and q of Mirror^d into another picture $[p/q]$ whose "top part", i.e. the half-picture over the diagonal, is the top part of p, and whose "down part" (half-picture under the diagonal) is the down part of q. By definition, $[p/q]$ does not belong to Mirror^d. However, a locality and counting argument yields that it *should belong* to this language, a contradiction. In order to detail this argument, we introduce some preliminary notations.

Given $n, d > 0$, we define the three following subsets of $[n]^d$:

1. $top([n]^d) = \{\mathbf{a} \in [n]^d : [\mathbf{a}]_1 \leq [\mathbf{a}]_2\}$;

2. *down*($[n]^d$) = $\{\mathbf{a} \in [n]^d : [\mathbf{a}]_1 > [\mathbf{a}]_2\}$;

3. *diag*($[n]^d$) = $\{\mathbf{a} \in [n]^d : [\mathbf{a}]_1 = [\mathbf{a}]_2 \text{ or } [\mathbf{a}]_1 = [\mathbf{a}]_2 + 1\}$.

Thus, top($[n]^d$) contains the cells over the diagonal, down($[n]^d$), the cells strictly under the diagonal, and diag($[n]^d$), the cells lying on the first or the second diagonal.

4. For a picture $p : [n]^d \to \Sigma$, we denote *diag*(p) the restriction of p to diag($[n]^d$).

5. For a picture language L, we set *diag*(L) = $\{\text{diag}(p) : p \in L\}$.

6. Given two d-pictures $p, q : [n]^d \to \Sigma$, we denote by $[p/q]$ the picture defined as follows: $[p/q]$ has prototype $[n]^d \to \Sigma$ and for all $\mathbf{a} \in [n]^d$,

$$[p/q](\mathbf{a}) = \begin{cases} p(\mathbf{a}) \text{ if } [\mathbf{a}]_1 \leq [\mathbf{a}]_2; \\ q(\mathbf{a}) \text{ otherwise.} \end{cases}$$

In other words, $[p/q]$ is the picture that coincides with p on its top part and with q on its down part.

7. Given a d-language L we denote $L_n = \{p \in L : \text{dom}(p) = [n]^d\}$.

Let's now come back to the Mirror language. We have assumed that Mirror^d is the projection of a local language over Γ: $\text{Mirror}^d = \pi(\text{Loc})$ with $\text{Loc} = L(\Delta_1, \ldots, \Delta_d)$. It immediately yields: $\text{Mirror}^d_n = \pi(\text{Loc}_n)$. It is easily seen that $|\text{Mirror}^d_n| \geq |\Sigma|^{cn^d}$ for some constant c (take $c = 1/2$ for instance). Beside, $|\text{diag}(\text{Loc}_n)| \leq |\Gamma|^{2n^{d-1}}$ since for each n, $|\text{diag}([n]^d)| = 2n^{d-1}$. As $|\Sigma|^{cn^d} > |\Gamma|^{2n^{d-1}}$ for sufficiently large n, we have for some n: $|\text{Mirror}^d_n| > |\text{diag}(\text{Loc}_n)|$.

By the *pigeonhole principle*, this last inequality guarantees the existence of two pictures $\pi(p), \pi(q) \in \text{Mirror}^d_n = \pi(\text{Loc}_n)$ (i.e. $p, q \in \text{Loc}$), such that:

(*i*) $\pi(p) \neq \pi(q)$;

(*ii*) $\text{diag}(p) = \text{diag}(q)$.

Items (*i*) and (*ii*) clearly force $\underline{[\pi(p)/\pi(q)] \notin \text{Mirror}^d}$. In the same time, we deduce from (*ii*) that $[p/q]$ is tiled by $(\Delta_1, \ldots, \Delta_d)$, as p and q are. This is because for every $j \in [d]$, any pair of j-adjacent points $\mathbf{a}, \mathbf{b} \in [n]^d$:

- either are both in top($[n]^d$), where $[p/q]$ coincides with p,
- or are both in down($[n]^d$), where $[p/q]$ coincides with q,
- or are both in diag($[n]^d$), where $[p/q]$ coincides both with p and q.

In all cases, the pattern $([p/q](\mathbf{a}), [p/q](\mathbf{b}))$ is either $(p(\mathbf{a}), p(\mathbf{b}))$ or $(q(\mathbf{a}), q(\mathbf{b}))$, hence belonging to Δ_j. Consequently, $[p/q] \in L(\Delta_1, \ldots, \Delta_d) = \text{Loc}$ and therefore, $\pi([p/q]) \in \pi(\text{Loc})$. But clearly, $\pi([p/q]) = [\pi(p)/\pi(q)]$, and by definition, $\pi(\text{Loc}) = \text{Mirror}^d$. Hence, $\underline{[\pi(p)/\pi(q)] \in \text{Mirror}^d}$: a contradiction. □

Proposition 9.7. *For each integer $d \geq 2$ and for d-languages* represented by their coordinate structures*, the following strict inclusions hold:*

$$\mathrm{ESO}(\mathrm{var}\, d-1) \subsetneq \mathrm{REC}^d \subsetneq \mathrm{ESO}(\mathrm{var}\, d).$$

That means the following implications hold,

$$\mathit{coord}^d(L) \in \mathrm{ESO}(\mathrm{var}\, d-1) \Rightarrow L \in \mathrm{REC}^d \Rightarrow \mathit{coord}^d(L) \in \mathrm{ESO}(\mathrm{var}\, d),$$

but neither of their converses does.

Proof.

1. $\mathit{coord}^d(L) \in \mathrm{ESO}(\mathrm{var}\,(d-1)) \Rightarrow L \in \mathrm{REC}^d$. This implication is a direct consequence of the equivalence $L \in \mathrm{REC}^d \Leftrightarrow \mathsf{coord}^d(L) \in \mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \mathrm{sorted})$ stated in Fact 9.4 and of the following equalities/inclusions, that hold on coordinate pictures of dimension d:

$$\mathrm{ESO}(\mathrm{var}\,(d-1)) = \mathrm{ESO}(\forall^{d-1}, \mathrm{arity}\,(d-1)) \subseteq \mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \mathrm{sorted}).$$

The equality has been proved in Theorem 5.1 ; the inclusion can be proved exactly as the inclusion stated in Proposition 8.6, that is: $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \mathrm{half\text{-}sorted}) \subseteq \mathrm{ESO}(\forall^d, \mathrm{arity}\, d, \mathrm{sorted})$.

Furthermore, the converse implication $L \in \mathrm{REC}^d \Rightarrow \mathsf{coord}^d(L) \in \mathrm{ESO}(\mathrm{var}\,(d-1))$ does not hold. To see this, consider the set L_1 of d-pictures p on $\Sigma = \{0, 1\}$ whose pixels all contain symbol 1. Trivially, $L_1 \in \mathrm{REC}^d$. However,

$$\mathsf{coord}^d(L_1) \notin \mathrm{ESO}(\mathrm{var}\,(d-1)).$$

Let us justify this last assertion. Intuitively, each sentence in $\mathrm{ESO}(\mathrm{var}\,(d-1)) = \mathrm{ESO}(\forall^{d-1}, \mathrm{arity}\,(d-1))$ can only express constraints about $O(n^{d-1})$ pixels. More precisely, for any sentence $\varphi \in \mathrm{ESO}(\forall^{d-1}, \mathrm{arity}\,(d-1))$ over the d-coordinate Σ-signature, let the integer k be the maximal composition depth of the successor function in φ. Any atom of φ that involves an input relation symbol $Q_s, s \in \Sigma$, is of the form $Q_s(\tau_1(x_{i_1}), \ldots, \tau_d(x_{i_d}))$, for d terms $\tau_j(x_{i_j})$, $i_j \in [d-1]$. By the *pigeonhole principle*, at least two of those terms involve the same variable x_i, i.e., there are two distinct indices $1 \leq j < j' \leq d$, such that $i_j = i_{j'}$. For any d-picture $p : [n]^d \to \Sigma$, let $D_k(n)$ denote the set of pixels $\mathbf{a} = (a_1, \ldots, a_d) \in [n]^d$ for which there are two coordinates $a_j, a_{j'}$, $1 \leq j < j' \leq d$, at distance at most k: $|a_j - a_{j'}| \leq k$. Of course, $|D_k(n)| = O(n^{d-1})$. It is easy to convince oneself that the sentence φ cannot control any pixel of p outside the $O(n^{d-1})$ pixels of $D_k(n)$. That means that if two pictures $p, p' : [n]^d \to \Sigma$ coincide on those pixels, i.e. if $p(\mathbf{a}) = p'(\mathbf{a})$ for every $\mathbf{a} \in D_k(n)$, then $\mathsf{coord}^d(p) \models \varphi \Leftrightarrow \mathsf{coord}^d(p') \models \varphi$. Clearly, such a sentence cannot define $\mathsf{coord}^d(L_1)$.

2. $L \in \mathrm{REC}^d \Rightarrow \mathit{coord}^d(L) \in \mathrm{ESO}(\mathrm{var}\, d)$. This implication is a weakening of the equivalence stated in Fact 9.4. However, Lemma 9.6 implies that the converse implication is false. It completes the proof. ◻

For pictures languages of dimension $d \geq 2$ and coordinate encoding, the definability class $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$, although logically robust, does not coincide with some complexity class. However, the additional expressivity power due to the ability to freely permute the coordinates (i.e. the arguments of relation symbols) in a sentence of this logic intuitively corresponds to the ability to add in the neighborhood of any pixel $\mathbf{a} = (a_1, \ldots, a_d) \in [n]^d$ all its permuted pixels $\mathbf{a}_\alpha = (a_{\alpha(1)}, \ldots, a_{\alpha(d)})$, for each permutation α of $[d]$. This remark will be made precise using the notion of *folded picture*.

9.2.3. Folding: a means to force coordinate representation to fit recognizable picture languages

Definition 9.8. *For any integer $d \geq 1$, recall that $\mathcal{S}(d)$ denote the set of permutations of $[d]$. Besides, given an alphabet Σ we write Σ_d for the set of functions of prototype $\mathcal{S}(d) \to \Sigma$, and we set $\Sigma_d^0 = \Sigma_d \cup \{0\}$. Finally, we call folding the mapping that maps any d-picture $p : [n]^d \to \Sigma$ to its folded picture*

$$p^{\mathsf{fold}} : [n]^d \to \Sigma_d^0$$

defined as follows: for all $\mathbf{a} \in [n]^d$,

1. *if $\mathbf{a}$ is* increasing, *then $p^{\mathsf{fold}}(\mathbf{a})$ is the function of Σ_d that maps α on $p(\mathbf{a}_\alpha)$ (i.e. $p^{\mathsf{fold}}(\mathbf{a})(\alpha) = p(\mathbf{a}_\alpha)$);*
2. *otherwise, $p^{\mathsf{fold}}(\mathbf{a}) = 0$.*

(Remember that $\mathbf{a}$ is increasing, denoted by $\mathbf{a}\uparrow$, if $a_1 \leq a_2 \leq \cdots \leq a_d$. See Definition 6.2 (2))

Notice that $p \mapsto p^{\mathsf{fold}}$ is a *one-one* mapping from the set of d-pictures on Σ to the set of d-pictures on Σ_d^0 that fulfill condition 2: $p^{\mathsf{fold}}(\mathbf{a}) = 0$ if not $\mathbf{a}\uparrow$. All the information about p is contained in the part of p^{fold} that consists of increasing pixels. For example, for $d = 2$, if $a_1 \leq a_2$, then the "folded point" $p^{\mathsf{fold}}(a_1, a_2)$ contains both $p(a_1, a_2)$ and $p(a_2, a_1)$, and $p^{\mathsf{fold}}(a_2, a_1) = 0$. In other terms, all the information on p is gathered over the diagonal $a_1 = a_2$, i.e. on the pixels (a_1, a_2) of p^{fold} such that $a_1 \leq a_2$.

Definition 9.9. *For any d-language L on Σ, its folded language is the d-language on Σ_d^0:*

$$L^{\mathsf{fold}} = \{p^{\mathsf{fold}} : p \in L\}.$$

We now establish a simple correspondence between the coordinate encodings of p^{fold} and p.

Lemma 9.10. *For any d-picture p on Σ, any* increasing *point $\mathbf{a} \in dom(p)$, and every $\sigma \in \Sigma_d$,*

$$\mathsf{coord}^d(p^{\mathsf{fold}}) \models Q_\sigma(\mathbf{a}) \Leftrightarrow \mathsf{coord}^d(p) \models \bigwedge_{\alpha \in \mathcal{S}(d)} Q_{\sigma(\alpha)}(\mathbf{a}_\alpha).$$

Proof. Definitions 9.8 and 9.9 yield the following equivalences: for any d-picture p on Σ, any increasing point $\mathbf{a} \in dom(p)$, and any $\sigma \in \Sigma_d$,

$$\mathsf{coord}^d(p^{\mathsf{fold}}) \models Q_\sigma(\mathbf{a}) \Leftrightarrow p^{\mathsf{fold}}(\mathbf{a}) = \sigma \Leftrightarrow \bigwedge_{\alpha \in \mathcal{S}(d)} p^{\mathsf{fold}}(\mathbf{a})(\alpha) = \sigma(\alpha).$$

Beside, for each $\alpha \in \mathcal{S}(d)$, $p^{\mathsf{fold}}(\mathbf{a})(\alpha) = \sigma(\alpha) \Leftrightarrow p(\mathbf{a}_\alpha) = \sigma(\alpha) \Leftrightarrow \mathsf{coord}^d(p) \models Q_{\sigma(\alpha)}(\mathbf{a}_\alpha)$. This leads to the claimed equivalence. □

The following proposition states the precise relationships between the logic $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$ and the recognizable d-languages.

Proposition 9.11. *For any $d \geq 1$ and any d-language L, the following assertions are equivalent:*

1. *$\mathsf{coord}^d(L) \in \mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$;*
2. *$L^{\mathsf{fold}} \in \mathrm{REC}^d$.*

Proof. $\boxed{1 \Rightarrow 2}$ By Fact 9.4, one has to prove the implication

$$\mathsf{coord}^d(L) \in \mathrm{ESO}(\forall^d, \text{arity } d) \Rightarrow \mathsf{coord}^d(L^{\mathsf{fold}}) \in \mathrm{ESO}(\forall^d, \text{arity } d, \text{sorted}). \tag{36}$$

The proof is similar to that of the statement of Proposition 7.7:

$$\mathrm{ESO}(\forall^d, \text{arity } d) \subseteq \mathrm{ESO}(\forall^d, \text{arity } d, \text{half-sorted}).$$

The two proofs involve the same tools and similar formulas. Therefore, we essentially justify here the additional points. Let L be a d-language on Σ whose set of coordinate structures $\mathsf{coord}^d(L)$ is defined by an $\mathrm{ESO}(\forall^d, \text{arity } d)$ sentence Φ, that is $\Phi = \exists \mathbf{R} \forall \mathbf{x}\ \varphi(\mathbf{x})$, where $\mathbf{R}$ is a set of relation variables of arity d and $\varphi(\mathbf{x})$ is a quantifier-free formula whose list of first-order variables is $\mathbf{x} = (x_1, \ldots, x_d)$. The main idea is the following. Instead of associating families of $d!$ relation symbols $(R_\alpha)_{\alpha \in \mathcal{S}(d)}$ with the sole *guessed* relation symbols $R \in \mathbf{R}$, we do the same for *input* relation symbols (also of arity d) Q_s, $s \in \Sigma$. Thus, denoting $\mathrm{Rel}(\varphi) = \{\mathbf{R}, (Q_s)_{s \in \Sigma}\}$, we associate with each $(R, \alpha) \in \mathrm{Rel}(\varphi) \times \mathcal{S}(d)$ a d-ary relation symbol R_α[6], according to Definition 7.4. That is, we set:

$$R_\alpha = \{\mathbf{a} \in [n]^d : \mathbf{a}\!\uparrow \textit{ and } \mathbf{a}_{\alpha^{-1}} \in R\}. \tag{37}$$

Let's denote by $\varphi^{\mathsf{fold}}(\mathbf{x})$ the conjunction of the following quantifier-free formulas:

- $\varphi_0(\mathbf{x}) = \neg\, (\mathbf{x}\!\uparrow) \rightarrow Q_0(\mathbf{x})$;
- $\varphi_1(\mathbf{x}) = \displaystyle\bigwedge_{R \in \mathrm{Rel}(\varphi)} \bigwedge_{\alpha \in \mathcal{S}(d)} (R_\alpha(\mathbf{x}) \ \rightarrow \mathbf{x}\!\uparrow)$;
- $\varphi_2(\mathbf{x}) = \displaystyle\bigwedge_{R \in \mathrm{Rel}(\varphi)} \bigwedge_{\alpha \in \mathcal{S}(d)} \bigwedge_{\tau \in \mathcal{T}(d)} (\mathbf{x}_\tau = \mathbf{x} \rightarrow (R_\alpha(\mathbf{x}) \leftrightarrow R_{\alpha\tau}(\mathbf{x})))$;
- $\varphi_3(\mathbf{x}) = \mathbf{x}\!\uparrow\rightarrow \displaystyle\bigwedge_{\sigma \in \Sigma_d} \left(Q_\sigma(\mathbf{x}) \leftrightarrow \bigwedge_{\alpha \in \mathcal{S}(d)} Q_{\sigma(\alpha), \alpha^{-1}}(\mathbf{x}) \right)$;
- $\tilde{\varphi}(\mathbf{x}) = \displaystyle\bigwedge_{\alpha \in \mathcal{S}(d)} \Big(\mathbf{x}\!\uparrow\rightarrow \tilde{\varphi}_\alpha(\mathbf{x}, (R_\gamma)_{\gamma \in \mathcal{S}(d)})\Big)$.

where $\tilde{\varphi}_\alpha$ is obtained from φ by substituting R-atoms, for $R \in \mathrm{Rel}(\varphi)$, with some "sorted" R_γ-atoms $R_\gamma(\mathbf{x})$ or $R_\gamma(\mathbf{x}^{(j)})$, $j \in [d]$, see Lemma 7.6. Let us now define the $\mathrm{ESO}(\forall^d, \text{arity } d, \text{sorted})$-sentence

$$\Phi^{\mathsf{fold}} = \exists (R_\alpha)_{R \in \mathrm{Rel}(\varphi),\ \alpha \in \mathcal{S}(d)}\ \forall \mathbf{x}\ \varphi^{\mathsf{fold}}(\mathbf{x}).$$

We claim that for every d-picture p on Σ, the following equivalence holds:

$$\mathsf{coord}^d(p) \models \Phi \Leftrightarrow \mathsf{coord}^d(p^{\mathsf{fold}}) \models \Phi^{\mathsf{fold}}. \tag{38}$$

In this purpose, let us comment upon the above conjuncts φ_0–φ_3. Formulas φ_0–φ_2 correspond to those displayed in Equation (25) of Proposition 7.7. Clearly, φ_0 and φ_1 express that in the folded picture p^{fold}, all the informations are gathered in the "folded part" (the increasing pixels) while the other pixels contain 0.

[6] For Q_s, it is also denoted $Q_{s,\alpha}$.

Formula φ_2 ensures that for each $R \in \mathrm{Rel}(\varphi)$, its associated relations R_α's agree on their common parts as mentioned in Section 7. Moreover, φ_3 expresses that the relations Q_σ, $\sigma \in \Sigma_d$, are what they should be for the folded picture p^{fold}, as explained here below. By (37) applied to the relation $Q_{\sigma(\alpha),\alpha^{-1}}(\mathbf{a})$ associated with $Q_{\sigma(\alpha)}$, we have:

$$Q_{\sigma(\alpha),\alpha^{-1}} = \{\mathbf{a} \in [n]^d : \mathbf{a}\!\uparrow \ \text{and} \ \ \mathbf{a}_\alpha \in Q_{\sigma(\alpha)}\}.$$

Hence, by Lemma 9.10,

$$Q_{\sigma(\alpha),\alpha^{-1}} = \{\mathbf{a} \in [n]^d : \mathbf{a}\!\uparrow \ \text{and} \ \ \mathbf{a} \in Q_{\sigma}\}.$$

This corresponds to equivalence $Q_\sigma(\mathbf{x}) \leftrightarrow (\bigwedge_{\alpha\in\mathcal{S}(d)} Q_{\sigma(\alpha),\alpha^{-1}}(\mathbf{x})$ of φ_3. Finally, Equivalence (38) is justified similarly as one shows that $\mathrm{ESO}(\forall^d, \text{arity}\, d) = \mathrm{ESO}(\forall^d, \text{arity}\, d, \text{half-sorted}\,)$, in Proposition 7.7. In order to prove the implication (36) and hence implication 1 $\Rightarrow$ 2, it is sufficient to justify the following equivalence, for every d-picture p' on Σ_d^0:

$$\mathsf{coord}^d(p') \models \Phi^{\mathsf{fold}} \Leftrightarrow \exists p\,(p' = p^{\mathsf{fold}} \text{ and } \mathsf{coord}^d(p) \models \Phi).$$

In view of equivalence (38), there only remains to justify the following implication:

$$\mathsf{coord}^d(p') \models \Phi^{\mathsf{fold}} \Rightarrow \exists p(p' = p^{\mathsf{fold}}).$$

It is clear that the "subformula" $\forall\mathbf{x} : \ \varphi_0(\mathbf{x}) \wedge \varphi_1(\mathbf{x}) \wedge \varphi_2(\mathbf{x}) \wedge \varphi_3(\mathbf{x})$ of Φ^{fold} implies that the underlying picture is of the form p^{fold} as required. This concludes the proof of implication 1 $\Rightarrow$ 2. Let's now prove the converse one.

$\boxed{2 \Rightarrow 1}$ Assume that the folded language L^{fold} of a d-language L on Σ is recognizable. By Fact 9.4 the set of coordinate structures $\mathsf{coord}^d(L^{\mathsf{fold}})$ is defined in $\mathrm{ESO}(\forall^d, \text{arity}\, d)$, i.e., by a sentence Φ of the form $\Phi = \exists\mathbf{R}\forall\mathbf{x}\ \varphi(\mathbf{x})$, where $\mathbf{R}$ is a set of d-ary relation variables and φ is a quantifier-free formula with the list of first-order variables $\mathbf{x} = (x_1, \dots, x_d)$. The intuitive idea of the construction of the following sentence, denoted Φ^{unfold}, also in $\mathrm{ESO}(\forall^d, \text{arity}\, d)$, that defines the set of coordinate structures $\mathsf{coord}^d(L)$, is to view each picture $p \in L$ as the projection of a picture that is the *superposition* of p and of its folded picture p^{fold}:

$$\Phi^{\mathsf{unfold}} = \exists(Q_\sigma)_{\sigma\in\Sigma_d^0}\exists\mathbf{R}\forall\mathbf{x}\ \varphi^{\mathsf{unfold}}(\mathbf{x})$$

where

$$\varphi^{\mathsf{unfold}}(\mathbf{x}) = \varphi(\mathbf{x}) \wedge \left\{ \begin{array}{ll} & (\mathbf{x}\!\uparrow) \rightarrow \displaystyle\bigwedge_{\sigma\in\Sigma_d} (\,Q_\sigma(\mathbf{x}) \leftrightarrow \bigwedge_{\alpha\in\mathcal{S}(d)} Q_{\sigma(\alpha)}(\mathbf{x}_\alpha)\,) \\ \wedge & \neg(\mathbf{x}\!\uparrow) \rightarrow (\,Q_0(\mathbf{x}) \wedge \displaystyle\bigwedge_{\sigma\in\Sigma_d} \neg Q_\sigma(\mathbf{x})\,). \end{array} \right\}$$

In order to justify the correctness of sentence Φ^{unfold}, that means, for each d-picture p on Σ,

$$p \in L \Leftrightarrow \mathsf{coord}^d(p) \models \Phi^{\mathsf{unfold}},$$

it is sufficient to prove the following equivalence:

$$\mathsf{coord}^d(p^{\mathsf{fold}}) \models \Phi \Leftrightarrow \mathsf{coord}^d(p) \models \Phi^{\mathsf{unfold}}. \tag{39}$$

The implication $\Leftarrow$ of (39) is obvious since φ is a conjunct of $\varphi^{\mathsf{unfold}}$. The converse implication $\Rightarrow$ is a straigtforward consequence of the definitions and, for conjunct 9.2.3, of Lemma 9.10. Proposition 9.11 is proved. $\square$

10. Hierarchy results

We are now ready to prove some strict hierarchy results between the classes REC^d, $\mathrm{NLIN}^d_{\mathrm{ca}}$ and our logical classes.

For that purpose we will use the notion of "folded" language with Proposition **??** and the following "symmetric" language as a counterexample.

Definition 10.1. *Let $\textsc{sym}_d$ be the d-language on $\Sigma = \{0,1\}$ defined as follows: a d-picture $p : [n]^d \to \{0,1\}$ belongs to $\textsc{sym}_d$ iff, for all $\mathbf{a} = (a_1, \ldots, a_d) \in [n]^d$, we have:*

1. *$p(\mathbf{a}) = p(\mathbf{a}_\alpha)$, for all permutation $\alpha \in \mathcal{S}(d)$, where $\mathbf{a}_\alpha = (a_{\alpha(1)}, \ldots, a_{\alpha(d)})$;*
2. *$p(\mathbf{a}) = p(\mathbf{a}_{sym(i)})$, for all $i \in [d]$, where $\mathbf{a}_{sym(i)}$ denotes the tuple $\mathbf{a}$ whose i^{th} component a_i is replaced by its "symmetric value" $n+1-a_i$[7].*

In other words, the values $p(\mathbf{a})$ are defined up to all possible permutations of coordinates and up to all possible symmetries $\mathbf{a} \mapsto \mathbf{a}_{sym(i)}$.

There is another equivalent definition of language $\textsc{sym}_d$ that uses an equivalence relation on pixels.

Definition 10.2. *Let $\sim_n$ be the equivalence relation on $[n]^d$ defined as follows: for all $\mathbf{a} = (a_1, \ldots, a_d)$ and $\mathbf{b} = (b_1, \ldots, b_d)$ in $[n]^d$, $\mathbf{a} \sim_n \mathbf{b}$ holds iff there is some permutation $\alpha \in \mathcal{S}(d)$ such that, for each $i \in [d]$, $a_{\alpha(i)} = b_i$ or $a_{\alpha(i)} = n+1-b_i$.*

Lemma 10.3.
1. *A picture $p : [n]^d \to \{0,1\}$ is in $\textsc{sym}_d$ iff for all $\mathbf{a}, \mathbf{b} \in [n]^d$, $\mathbf{a} \sim_n \mathbf{b}$ implies $p(\mathbf{a}) = p(\mathbf{b})$.*
2. *For each $\mathbf{a} \in [n]^d$, there is exactly one d-tuple $\mathbf{b} = (b_1, \ldots, b_d)$ such that $\mathbf{a} \sim_n \mathbf{b}$ and $1 \le b_1 \le b_2 \le \cdots \le b_d \le (n+1)/2$. This tuple is called the representer of $\mathbf{a}$ in $[\lfloor (n+1)/2 \rfloor]^d$, denoted $\mathbf{b} = rep(\mathbf{a})$.*
3. *There is a bijective mapping $p \mapsto p'$ from the set of pictures p of $\textsc{sym}_d$ and of domain $[n]^d$ onto the set of functions $p' : Rep_n \to \{0,1\}$ where $Rep_n = \{\mathbf{b} \in [\lfloor (n+1)/2 \rfloor]^d : \mathbf{b}\uparrow\}$. That is, p' is the restriction of p to the subdomain Rep_n.*
4. *The number of pictures of $\textsc{sym}_d$ and of domain $[n]^d$ is at least 2^{cn^d}, for some constant c that depends on d.*

Proof. Assertion 1 is a straightforward consequence of Definitions 10.1 and 10.2. Assertion 2 easily follows from the following observation: for all integer $u \in [n]$, exactly one of the following conditions holds:

- $1 \le u < (n+1)/2$;
- $1 \le n+1-u < (n+1)/2$;
- $u = (n+1)/2 = n+1-u$.

[7]Obviously, because of item 1, item 2 is equivalent to the same assertion for only $i = 1$.

Assertion 3 comes easily from 1 and 2. Item 3 implies 4 because $|\mathrm{Rep}_n| \geq cn^d$, for some constant c that depends on d[8] and hence the number of functions from Rep_n to $\{0,1\}$ is at least 2^{cn^d}. ◻

The following lemma about the folded language $(\textsc{sym}_d)^{\mathsf{fold}}$ is easily deduced from Lemma 10.3 with the definition of folding.

Lemma 10.4. *1. The mapping $p \mapsto p'$ that maps any picture p of $(\textsc{sym}_d)^{\mathsf{fold}}$ and of domain $[n]^d$ to its restriction p' to the subdomain $Rep_n = \{a \in [\lfloor (n+1)/2 \rfloor]^d : \mathbf{a}\uparrow\}$ is one-one.*

2. *Also, the mapping $p \mapsto p''$ that maps any picture p of $(\textsc{sym}_d)^{\mathsf{fold}}$ and of domain $[n]^d$ to its restriction p'' to the subdomain $T_n = \{\mathbf{a} \in [n]^d : \mathbf{a}\uparrow$ and $\mathbf{a} \notin Rep_n\}$ is one-one.*

3. *The number of pictures of $(\textsc{sym}_d)^{\mathsf{fold}}$ and of domain $[n]^d$ is at least 2^{cn^d}, for some constant c that depends on d.*

Proof. Item 1 comes from item 3 of Lemma 10.3. Item 2 is justified similarly: it is sufficient to note that each point $\mathbf{a} \in [n]^d$ has at least one equivalent element $\mathbf{a} \sim_n \mathbf{b}$ in T_n. (For $d \geq 2$, there are in fact at least two such equivalent elements in T_n.) Item 3 is deduced from item 4 of Lemma 10.3. ◻

Lemma 10.5. *For all $d \geq 2$, we have $\mathsf{coord}^d(\textsc{sym}_d) \notin \mathrm{ESO}(\forall^d, \mathrm{arity}\ d)$.*

Proof. By Proposition 9.11, it is sufficient to prove $L \notin \mathrm{REC}^d$ for the folded language $L = (\textsc{sym}_d)^{\mathsf{fold}}$. From Lemma 10.3 for $\textsc{sym}_d$ and the definition of $L = (\textsc{sym}_d)^{\mathsf{fold}}$, one easily deduces that L is the d-language on Σ_d^0, where $\Sigma = \{0,1\}$, that is defined as follows: a d-picture p on Σ_d^0 belongs to L iff the following two clauses hold, for all $\mathbf{a} \in [n]^d$:

1. if $\mathbf{a}\uparrow$ then $p(\mathbf{a})$ is a *constant mapping*[9] $p(\mathbf{a}) : \mathcal{S}(d) \to \{0,1\}$ such that $p(\mathbf{a}) = p(\mathrm{rep}(\mathbf{a}))$, where $\mathrm{rep}(\mathbf{a})$ is the representer of $\mathbf{a}$ in $[\lfloor (n+1)/2 \rfloor]^d$, i.e. the unique tuple $\mathbf{b} = (b_1, \ldots, b_d)$ such that $\mathbf{a} \sim_n \mathbf{b}$ and $1 \leq b_1 \leq b_2 \leq \cdots \leq b_d \leq (n+1)/2$;
2. otherwise, $p(\mathbf{a}) = 0$.

For simplicity (abuse) of notation, in case $\mathbf{a}\uparrow$, i.e., when $p(\mathbf{a})$ is a constant mapping, the constant value $p(\mathbf{a})(\alpha) \in \{0,1\}$ for all $\alpha \in \mathcal{S}(d)$ is also denoted by $p(\mathbf{a})$. So, L will be regarded as a d-language on alphabet $\Sigma = \{0,1\}$ (instead of alphabet Σ_d^0) defined by clauses 1 and 2 above. In other words, the expression "$p(\mathbf{a})$ is a constant mapping from $\mathcal{S}(d)$ to $\{0,1\}$" is replaced in clause 1 by "$p(\mathbf{a}) \in \{0,1\}$". We have to prove that L is not recognizable. Our proof by contradiction is very similar to the proof of the same result for the d-language Mirror^d (see Lemma 9.6).

Assume that L is recognizable, that is, L is the projection of some local d-language L' on Γ by some function $\pi : \Gamma \to \{0,1\}$. Let us introduce the following notation: for two pictures, $p_1, p_2 : [n]^d \to \Gamma$, let $p_{\mathsf{mix}} = [p_2/p_1]$ denote the picture $p_{\mathsf{mix}} : [n]^d \to \Gamma$ defined as follows: for all $\mathbf{a} \in [n]^d$,

1. "down" for $\mathbf{a}\uparrow$: $p_{\mathsf{mix}}(\mathbf{a}) = p_1(\mathbf{a})$ if $\mathbf{a} \in \mathrm{Rep}_n$, i.e., $\mathbf{a}\uparrow$ and $\mathbf{a} \in [\lfloor (n+1)/2 \rfloor]^d$;
2. "top" for $\mathbf{a}\uparrow$: $p_{\mathsf{mix}}(\mathbf{a}) = p_2(\mathbf{a})$ if $\mathbf{a} \in T_n$, i.e, $\mathbf{a}\uparrow$ and $\mathbf{a} \notin \mathrm{Rep}_n$;

[8] For example, for $d = 2$, Rep_n is the set of ordered pairs of integers (a_1, a_2) such that $1 \leq a_1 \leq a_2 \leq (n+1)/2$ and its cardinality is at least $1/2(n/2)^2 = 1/8\,n^2$.

[9] The fact that $p(\mathbf{a})$ should be a constant function is a straightforward consequence of item 1 in the above Definition 10.1 of $\textsc{sym}_d$.

3. Elsewhere: $p_{\mathsf{mix}}(\mathbf{a}) = 0$ otherwise, i.e., if not $\mathbf{a}\!\uparrow$.

For a positive integer n, let $L_n = \{p \in L : dom(p) = [n]^d\}$. Notice that the number of pictures in L_n is greater than 2^{cn^d}, for some constant c, by item 3 of Lemma 10.4. Let L'_n be the set of pictures in L' (on Γ) whose projections by π are in L_n. The restriction of a picture $p' : [n]^d \to \Gamma$ to the "border set"[10]

$$B_n = \{\mathbf{a} \in [n]^d : \mathbf{a}\!\uparrow \; and \; (a_d = \lfloor (n+1)/2 \rfloor \; or \; a_d = \lfloor (n+1)/2 \rfloor + 1)\}$$

is called the *border stripe* of p'. Let S_n be the set of border stripes of pictures of L'_n. Clearly, $|B_n| \leq 2n^{d-1}$ and then, $|S_n| \leq |\Gamma|^{2n^{d-1}}$. For n sufficiently large, we have $2^{cn^d} > |\Gamma|^{2n^{d-1}}$. Therefore, by the *pigeonhole principle*, for n sufficiently large, there will be two different pictures p_1 and p_2 in L_n whose corresponding pictures p'_1 and p'_2 in L'_n such that $p_1 = \pi \circ p'_1$ and $p_2 = \pi \circ p'_2$ have the same border stripe. By definition of a local language, that implies that the mixed picture

$$p'_{\mathsf{mix}} = [p'_2/p'_1]$$

also belongs to L'_n and therefore its projection

$$p_{\mathsf{mix}} = \pi \circ p'_{\mathsf{mix}} = [\pi \circ p'_2/\pi \circ p'_1] = [p_2/p_1]$$

belongs to L_n, i.e., is in $L = (\textsc{sym}_d)^{\mathsf{fold}}$ as p_1 an p_2 are. Now, notice that by clause 1 (resp. clause 2) of the definition of the picture $p_{\mathsf{mix}} = [p_2/p_1]$, the restriction of p_{mix} to the subdomain Rep_n (resp. T_n) *is* the restriction of p_1 to Rep_n (resp. the restriction of p_2 to T_n). This implies $p_{\mathsf{mix}} = p_1$ (resp. $p_{\mathsf{mix}} = p_2$) because this restriction is a one-one mapping as item 1 (resp. item 2) of Lemma 10.4 asserts. This is a contradiction since pictures p_1 and p_2 are different each other. This proves $L = (\textsc{sym}_d)^{\mathsf{fold}} \notin \mathrm{REC}^d$ and hence Lemma 10.5. □

We now have all the tools to prove the following strict hierarchy theorem.

Theorem 10.6. *For each integer $d \geq 2$ and for d-languages represented by* coordinate *structures, the following (strict) inclusions hold:*

$$\begin{array}{ccccc}
\mathrm{REC}^d & \subsetneq & \mathrm{ESO}(\mathrm{var}\, d) & = & \mathrm{ESO}(\forall^d, \mathrm{arity}\, d) \\
 & & & & \rotatebox{90}{$\subsetneq$} \\
 & & & & \mathrm{ESO}(\forall^{d+1}, \mathrm{arity}\, d) \\
 & & & & \rotatebox{90}{$\subseteq$} \\
\mathrm{NLIN}^d_{ca} & = & \mathrm{ESO}(\mathrm{var}\, d+1) & = & \mathrm{ESO}(\forall^{d+1}, \mathrm{arity}\, d+1)
\end{array}$$

Proof. Note that all the equalities and inclusions mentioned in the theorem either have been already proved or are trivial. There remains to justify the strictness of two of these inclusions. Proposition 9.7 states that the inclusion $\mathrm{REC}^d \subsetneq \mathrm{ESO}(\mathrm{var}\, d)$ is strict. Also, the inclusion $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d) \subsetneq \mathrm{ESO}(\forall^{d+1}, \mathrm{arity}\, d)$ is strict with the d-language $\textsc{sym}_d$ as a counterexample: Lemma 10.5 states that $\mathsf{coord}^d(\textsc{sym}_d) \notin \mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$, for all $d \geq 2$; so, it only remains to prove $\mathsf{coord}^d(\textsc{sym}_d) \in \mathrm{ESO}(\forall^{d+1}, \mathrm{arity}\, d)$. First, one can easily check that the first-order sentence $\forall x_1 \forall x_2 : \psi_0(x_1, x_2) \wedge \psi_1(x_1, x_2)$, where:

$$\begin{array}{lcll}
\psi_0 \equiv & \{\min(x_1) \vee \max(x_2))\} & \rightarrow & \left\{ R_{sym}(x_1, x_2) \leftrightarrow (\min(x_1) \vee \max(x_2)) \right\} \text{ and} \\
\psi_1 \equiv & \{\neg\max(x_1) \wedge \neg\max(x_2)\} & \rightarrow & \left\{ R_{sym}(\mathrm{succ}(x_1), x_2) \leftrightarrow R_{sym}(x_1, \mathrm{succ}(x_2)) \right\},
\end{array}$$

[10] B_n consists of two rows of pixels, the first one satisfying $a_d = \lfloor (n+1)/2 \rfloor$ *and* $\mathbf{a}\!\uparrow$ in Rep_n and the second one satisfying $a_d = \lfloor (n+1)/2 \rfloor + 1$ *and* $\mathbf{a}\!\uparrow$ in T_n.

defines the binary relation $R_{sym}(x_1,x_2)$ to be $x_1+x_2=n+1$ on any coordinate structure of domain $[n]$. Hence, by Definition 10.1 and its footnote, one easily sees that the set of structures $\mathsf{coord}^d(\textsc{sym}_d)$ is defined by the following sentence φ_{sym} in $\mathrm{ESO}(\forall^{d+1}, \mathrm{arity}\ d)$:

$$\exists R_{sym} \forall x_0 \ldots \forall x_d \left\{ \begin{array}{l} \psi_0(x_1,x_2) \ \wedge \ \psi_1(x_1,x_2) \ \wedge \\ \bigwedge_{\alpha \in \mathcal{S}(d)} (Q_1(\mathbf{x}) \leftrightarrow Q_1(\mathbf{x}_\alpha)) \ \wedge \\ R_{sym}(x_1,x_0) \to (Q_1(\mathbf{x}) \leftrightarrow Q_1(\mathbf{x}_{1\to 0})) . \end{array} \right\}$$

Here, as usual, $\mathbf{x}=(x_1,\ldots,x_d)$, $\mathbf{x}_\alpha=(x_{\alpha(1)},\ldots,x_{\alpha(d)})$, and $\mathbf{x}_{1\to 0}$ is the d-tuple $\mathbf{x}$ where x_1 is replaced by x_0. This achieves the proof of Theorem 10.6. □

The above strict inclusion $\mathrm{ESO}(\mathrm{var}\ d) \subsetneq \mathrm{ESO}(\forall^{d+1}, \mathrm{arity}\ d)$ trivially yields the following result.

Corollary 10.7. *For each integer $d \geq 2$ and for d-languages represented by* coordinate *structures, we have the* strict inclusion*:* $\mathrm{ESO}(\mathrm{var}\ d) \subsetneq \mathrm{ESO}(\mathrm{arity}\ d)$.

Remark 10.8. *Corollary 10.7 surprisingly contrasts with the equality* $\mathrm{ESO}(\mathrm{var}\ 1) = \mathrm{ESO}(\mathrm{arity}\ 1)$ *for d-languages represented by* pixels *structures, see Theorem 3.6 and Corollary 3.16.*

11. Conclusion: final remarks, additional results and open problems

11.1. Locality in logic

As recalled in the introduction of this paper there are several notions of locality in logic. The locality of general first-order definability expressed by the normal form of Gaifman's Theorem [19] is weaker than the locality of first-order logic definability *with only one variable* (universally quantified) *over picture languages* (with adjacency represented by successor functions), a notion that Borchert [2] has shown to be equivalent to tilability. Note that this strong locality is obtained *in accordance* with the locality of the underlying grid structure.

In this paper, we have used exclusively this stronger notion. We have established natural EMSO or ESO characterizations of two nondeterministic classes of picture languages: recognizable picture languages and linear time class of nondeterministic cellular automata.

Of course, those complexity/logical notions are nonlocal: concerning the most restrictive one, recognizable picture properties, surprising, Reinhardt [56, 57] has proved that several global properties including connectivity are recognizable and there are also NP-complete problems among recognizable ones.

However, the intuitive idea that both are classes of "projections" of local languages is made explicit by their characterization by ESO logics with normal forms *whose first-order part is local:* that means with only one first-order variable for pixel encoding, or – for coordinate encoding of d-pictures – with $d+1$ "sorted" first-order variables, one of which intuitively represents the time.

11.2. Extensions and limits of our results

It is interesting to notice that several of our results can be extended:

- Our logical normalization
$$\mathrm{ESO}(\mathrm{var}\ k) = \mathrm{ESO}(\forall^k, \mathrm{arity}\ k)$$
holds for sets of structures of *any arity d* provided they are equipped with a *successor function* (explicitly given or implicitly defined), in particular the sets $\mathsf{coord}^d(L)$ or $\mathsf{pixel}^d(L)$, for a picture language L of *any dimension d*.

- Our main characterization result of d-picture languages with coordinate encoding

$$\mathsf{coord}^d(L) \in \mathrm{ESO}(\mathrm{var}\, d+1) \Leftrightarrow L \in \mathrm{NLIN}^d_{\mathrm{ca}}$$

also holds (with the same proofs) for all dimensions $k > d$; that means, for any d-picture language L and all $k > d$, the set of coordinate structures $\mathsf{coord}^d(L)$ is definable in $\mathrm{ESO}(\mathrm{var}\, k)$ iff L is recognized in time $O(n)$ by some nondeterministic cellular automaton of dimension $k-1$, i.e. of set of cells $[n]^{k-1}$.

- The previous result is essentially due to the normalization result $\mathrm{ESO}(\forall^k, \mathrm{arity}\, k) = \mathrm{ESO}(\forall^k, \mathrm{arity}\, k, \mathrm{sorted}\,)$ that holds for all structures of arity d equipped with a *successor function*, *provided* $k > d$, and is false for $k \leq d$, as we have shown for $k = d$.

11.3. Other related results of the literature

Borchert [2] has stated some results to be compared with our logical characterizations of recognizable languages and of linear time bounded complexity classes of multidimensional cellular automata although, paradoxically, his paper never mentions cellular automata. More precisely, Giammarresi [24] and/or Borchert [2] have studied a class of word languages (resp. 2-picture languages called *graph languages*) that Borchert has called COL^d or d-dimensionally colorable language, for any fixed integer $d \geq 1$ (resp. $COL^d_{\mathcal{G}}$, for any fixed integer $d \geq 2$). Among other equivalent characterizations, [24] and/or [2] defined COL^d (resp. $COL^d_{\mathcal{G}}$) as the class of word (resp. 2-picture) languages L for which there exists a recognizable d-picture language L' such that L is the set of *frontiers* (resp. *square frontiers*) of the pictures of L': a word w (resp. 2-picture f) is the *frontier* (resp. *square frontier*) of a d-picture $p : [n]^d \to \Sigma$ if w is the word (resp. f is the d-picture) written on the first "edge" (resp. first "face") of the d-dimensional colored "cube" p, that means $w = w_1 w_2 \ldots w_n$ with $w_i = p(i, 1, \ldots, 1)$, for each $i \leq n$ (f is $f : [n]^2 \to \Sigma$ with $f(i, j) = p(i, j, 1, \ldots, 1)$, for all $(i, j \in [n]^2)$.

Indeed, from these definitions the following fact is easily deduced:

Fact 11.1.
- *COL^1 is the class of regular word languages and $COL^2_{\mathcal{G}}$ is the class of recognizable 2-picture languages, respectively: $COL^1 = REC^1 = REG$ and $COL^2_{\mathcal{G}} = REC^2$.*
- *For any $d > 1$ (resp. $d > 2$), a word language (resp. 2-picture language) L belongs to COL^d (resp. $COL^d_{\mathcal{G}}$) iff there exists a nondeterministic $(d-1)$-dimensional cellular automaton that recognizes L in linear time.*

Then Borchert [2] states logical characterizations (Lemma 5.2 and Corollary 9.2(d) in [2]) that can be rephrased as follows:

1. On words, $COL^1 = REC^1 = REG = \mathrm{ESO}(\forall^1, \mathrm{arity}\, 1)$.
2. For $d > 1$, a word language belongs to COL^d, i.e. is recognized by some nondeterministic $(d-1)$-dimensional cellular automaton in linear time, iff it is definable in $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$.
3. For $d > 2$, a 2-picture language belongs to $COL^d_{\mathcal{G}}$, i.e. is recognized by some nondeterministic $(d-1)$-dimensional cellular automaton in linear time, iff it is definable in $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d)$ for coordinate representation.

However, the sketchy proofs of results 2 and 3 given by [2] have some drawbacks. The implications “definability” → “complexity”, i.e. the inclusions $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d) \subseteq COL^d$ for word languages, and $\mathrm{ESO}(\forall^d, \mathrm{arity}\, d) \subseteq COL^d_{\mathcal{G}}$ for 2-picture languages, are not correctly justified in [2]. As we have seen in our proofs of Proposition 8.6, Theorem 8.7, and Proposition 9.2, one should beforehand normalize each sentence by "sorting" its first-order variables.

11.4. Some remarks about the dimensions of pictures

For sake of simplicity and uniformity, we have chosen in this paper to restrict the presentation of results to "square" pictures, i.e. pictures of prototype $p : [n]^d \to \Sigma$. This may appear as a too strong requirement. In this section, we explain how our results about logical classes and their relationships with complexity classes REC^d and $\mathrm{NLIN}^d_{\mathrm{ca}}$ can be extended to the "most general" picture languages, i.e. to sets of d-pictures of prototype $p : [n_1] \times \cdots \times [n_d] \to \Sigma$. "Most general" means as much general as they make sense in the logical or complexity theoretical framework involved.

11.4.1. REC^d *and logical characterizations in pixel encodings.*

All our characterizations of REC^d for pixel encodings (see Section 3) hold for pictures of general prototype $p : [n_1] \times \cdots \times [n_d] \to \Sigma$, without any restriction, i.e. for all $n_i \geq 1$, $i \in [d]$. Moreover, our proofs also hold without change, except, of course, the references to integer n which should be replaced by integer n_i according to the involved dimension $i \in [d]$.

11.4.2. NLIN^d_{ca} *and logical characterizations in coordinate encodings.*

The definition of linear time complexity of d-automata is not clear for input pictures whose domain has the general form $[n_1] \times \cdots \times [n_d]$ without restriction. However, linear time, i.e. time $O(n)$, makes sense when the n_i are of the same order $\Theta(n)$, i.e. the pictures are "well-balanced". This justifies the following definition.

Definition 11.2. *A d-picture language L is* well-balanced *if there is some constant positive integer c such that the domain $[n_1] \times \cdots \times [n_d]$ of each picture $p \in L$ fulfills the condition: for $i \in [d]$, $n \leq c\, n_i$, where $n = max(n_1, \ldots, n_d)$ is called the* length *of p; we say that the picture p (resp. the picture language L) is* c-balanced*.*

It is straightforward to adapt our notion of linear time to well-balanced picture languages.

Definition 11.3. *A c-balanced d-picture language L belongs to* NLIN^d_{ca} *if there exist a d-automaton $\mathcal{A}$ and a linear function $T(n) = c_1\, n + c_2$ such that L is the set of c-balanced d-pictures p accepted by $\mathcal{A}$ in time $T(n)$, where n is the length of p.*

Remark 11.4. *These notions are justified by the following points:*

- *The "perimeter" of a c-balanced d-picture p of length n is $\Theta(n)$ and its size (called area, or volume, etc., according to its dimension d) is $|p| = \Theta(n^d)$.*
- *So, linear time means time* linear in the perimeter *of the d-picture p or, equivalently, in $|p|^{1/d}$.*

We can easily extend all our results about square d-languages to well-balanced d-languages. In order to reduce the well-balanced case to the "square" case we need some new definitions.

Definition 11.5. *With each d-picture p of length n, one associates its* squared d-picture*, denoted by $p^=$, of domain $[n]^d$ obtained by putting the new special symbol $\square$ in each additional cell. Formally,*

$$p^=(a) = \begin{cases} p(a) \text{ if } a \in dom(p); \\ \square \text{ otherwise.} \end{cases}$$

With any d-picture language L one associates its squared d-picture language*, $L^= = \{p^= : p \in L\}$.*

Remark 11.6. *Let c be a constant. If p is a c-balanced d-picture, then the size of its squared picture $p^=$ is $|p^=| = \Theta(|p|)$*

The following result is easy to prove.

Lemma 11.7. *Let L be a well-balanced d-picture language. Then $L \in \mathrm{NLIN}^d_{ca} \Leftrightarrow L^= \in \mathrm{NLIN}^d_{ca}$.*

Remark 11.8. *Notice that the nontrivial (right-to-left) implication of the previous lemma means that the d-automaton $\mathcal{A}_1$ that recognizes $L^=$ in linear time can be simulated by some d-automaton $\mathcal{A}_2$ that recognizes L in the same time (up to a constant factor) but with less space: the computation area consists of the cells of p instead of the cells of its squared version $p^=$. This is possible with the following trick: the fact that the i^{th} dimension n of $p^=$ is replaced by $n_i \geq n/c$ allows to "fold" the picture c times along its i^{th}dimension. All in all, each cell of p simulates (at most) c^d cells of $p^=$. This is performed by taking for the set of states of $\mathcal{A}_2$ the set Γ^{c^d} where Γ is the set of states of $\mathcal{A}_1$.*

Now, let us compare any well-balanced d-picture language L and its squared language $L^=$ from a logical point of view. The domain of the coordinate representation of a d-picture $p : [n_1] \times \cdots \times [n_d]$ is naturally $[n]$ where n is the length of p, i.e. $n = \max(n_1, \ldots, n_d)$. So, we define the coordinate representation of p as

$$\mathsf{coord}^d(p) = \mathsf{coord}^d(p^=).$$

So, as a trivial consequence, $\mathsf{coord}^d(L) = \mathsf{coord}^d(L^=)$. This justifies that the results of Sections 4-10 also hold in the extended case of well-balanced d-picture languages.

11.5. More hierarchies

It is also natural to address the question of the strictness or collapsing of the following "hierarchies" about second-order logic (SO) over picture languages :

1. Is there a strict hierarchy of SO or MSO according to the *second-order quantifier alternation*?
2. Is there a strict hierarchy of the classes ESO(arity d) and ESO($\forall^k$, arity d) according to the *number of* ESO *relation symbols*?

11.5.1. Question 1: Hierarchies for second-order quantifier alternation

It is well-known that the answer is negative for MSO on word languages and tree languages: on these classes of languages, MSO = EMSO holds. That is, the hierarchy collapses at it first level: see e.g. Chapter 7 of Libkin's book [45].

At the opposite, Matz, Schweikardt and Thomas [51, 60, 49, 48] have answered Question 1 positively for MSO over 2-dimensional picture languages: the quantifier alternation is strict at each level for MSO on 2-picture languages in pixel representation (and, as a consequence, is also strict over the class of finite graphs).

Their proof is essentially based on the fact that, for any positive integer k, there is a function $f_k : \mathbb{N} \to \mathbb{N}$ (defined as a fixed height tower of exponentials) such that the set of rectangular grids of format $n \times f_k(n)$ (i.e, of width n and length $f_k(n)$) can be defined by some Σ_k MSO sentence but cannot be defined by some Σ_{k-1} MSO sentence.

So, the hierarchy result essentially rests on the (more than exponential) imbalance between the two dimensions of the rectangular grid.

In view of this result a natural question arises: Is the MSO-alternation hierarchy strict for *square* picture languages (or equivalently, *well-balanced* picture languages)? The only thing we know is that the class EMSO = REC^d is not closed under complement: notice *non*-$\mathrm{Mirror}^d \in$ EMSO but we have proved $\mathrm{Mirror}^d \notin$ EMSO.

11.5.2. Question 2: Hierarchies for number of second-order quantifiers

The answer is totally known and uniform for picture languages of any dimension. In all cases, the hierarchy collapses at its first level. More precisely, Thomas [70] has established that every EMSO sentence over words is equivalent to a sentence whose monadic quantifier prefix consists of a single existential quantifier. Matz [50] has proved the same result over 2-pictures in the pixel representation. (This strongly contrasts with a result by Otto [53] who proves a strict hierarchy at each level for the number of monadic quantifiers in EMSO over graphs.)

The proof of Matz can be extended (with the same arguments and slight adaptations) to any dimension d, for both pixel and coordinate representations. In other words, in both representations, all the logical classes – essentially $\mathrm{ESO}(\text{arity } d)$ and $\mathrm{ESO}(\forall^k, \text{arity } d)$ – we have studied over picture languages of any dimension are not modified by the requirement there should be only *one* ESO relation symbol.

11.6. Logical characterizations of complexity classes for linear time or time bounds defined up to a constant factor

Notice that Theorem 9.1 that characterizes the *linear* time complexity class of nondeterministic *cellular automata* is very similar to the following result about time complexity $O(n^d)$, for any $d \geq 1$, of nondeterministic *RAMs*, by the present authors [28]:

$$\mathrm{NTIME}_{\mathrm{ram}}(n^d) = \mathrm{ESOF}(\mathrm{var}\, d) = \mathrm{ESOF}(\forall^d, \mathrm{arity}\, d).$$

The main difference is that this latter result involves the existential second-order logic *with functions* (ESOF) *instead of* or *in addition to relations* and holds in all kinds of structures without restriction: pictures, structures of any arity and any type, etc. It is also interesting and maybe surprising to notice that, in those results, the *time degree* d of a RAM computation plays the same role as the *dimension* $d+1$ of the time-space diagram of a linear time bounded computation for a d-dimensional cellular automaton.

Both results attest of the robustness of the time complexity classes $\mathrm{NTIME}_{\mathrm{ram}}(n^d)$ and $\mathrm{NLIN}^d_{\mathrm{ca}}$. They stress the significance of the RAM's as a sequential model, and of the cellular automata as a parallel model.

Also, note that such machine-independent characterizations of complexity classes for linear time or other time bounds defined up to a constant factor are rather rare in the literature: as other examples in finite model theory we only know the logical characterization of linear time of nondeterministic Turing machines by Lautemann et al. [42] and the algebraic characterization of linear time of deterministic RAM's by Grandjean and Schwentick [30].

Acknowledgments

We warmly thank Gaétan Richard for a number of stimulating discussions and for having obtained in collaboration with us several results of this paper, in particular Proposition 4.10 and Proposition 5.2. The first author is grateful to Véronique Terrier for her suggestions, specially about the implicit links between cellular automata and recognizable picture languages, and for the many references to the literature she gave us. We also thank Oliver Matz for several interesting discussions and informations about recognizable picture languages and Nicolas Bacquey for having pointed us a mistake in a previous proof of Proposition 4.10.

References

[1] R. Barbanchon and E. Grandjean. Local problems, planar local problems and linear time. In Julian C. Bradfield, editor, *CSL*, volume 2471 of *Lecture Notes in Computer Science*, pages 397–411. Springer, 2002.

[2] Bernd Borchert. Formal language characterizations of P, NP, and PSPACE. *J. of Automata, Languages and Combinatorics*, 13(3/4):161–183, 2008.

[3] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Springer, 482 pages, Universitext, 2001.

[4] J. R. Büchi. Weak second-order arithmetic and finite automata. *Z. Math. Logik Grundlagen Math.*, 6:66–92, 1960.

[5] Stephen A. Cook. A hierarchy for nondeterministic time complexity. In Patrick C. Fischer, H. Paul Zeiger, Jeffrey D. Ullman, and Arnold L. Rosenberg, editors, *STOC*, pages 187–192. ACM, 1972.

[6] B. Courcelle and J. Engelfriet. *Graph structure and monadic second-order logic, a language theoretic approach*. Cambridge University Press, to appear in 2012.

[7] M. Delorme and J. Mazoyer, editors. *Cellular automata: A parallel model*. Springer, 373 pages, Mathematics and Its Applications, 1998.

[8] A. Durand and E. Grandjean. First-order queries on structures of bounded degree are computable with constant delay. *ACM Trans. Comput. Log.*, 8(4), 2007.

[9] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer-Verlag, 1995.

[10] C.C. Elgot. Decision problem of finite automata design and related arithmetics. *Trans. Amer. Math. Society*, 98:21–51, 1961.

[11] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. In R. M. Karp, editor, *Complexity of Computation, SIAM-AMS Proceedings*, pages 43–73, 1974.

[12] R. Fagin. Monadic generalized spectra. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 21:89–96, 1975.

[13] R. Fagin. A spectrum hierarchy. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 21:123–134, 1975.

[14] R. Fagin. Finite-model theory–a personal perspective. *Theoretical Computer Science*, 116(1):3–31, 1993.

[15] R. Fagin. Comparing the power of monadic NP games. In D. Leivant, editor, *Logic and Computational Complexity*, Lecture Notes in Computer Science, Vol. 960, pages 414–425. Springer-Verlag, Berlin/New York, 1995.

[16] R. Fagin, L. Stockmeyer, and M. Y. Vardi. On monadic np vs. monadic co-np. *Information and Computation*, 120(1):78–92, July 1995.

[17] A. Ferreira and H. Reichel, editors. *STACS 2001, 18th Annual Symposium on Theoretical Aspects of Computer Science, Dresden, Germany, February 15-17, 2001, Proceedings*, volume 2010 of *Lecture Notes in Computer Science*. Springer, 2001.

[18] J. Flum, E. Grädel, and T. Wilke, editors. *Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas]*, volume 2 of *Texts in Logic and Games*. Amsterdam University Press, 2008.

[19] H. Gaifman. On local and nonlocal properties. In J. Stern, editor, *Logic Colloquium '81*, pages 105–135. North Holland, 1982.

[20] D. Giammarresi and A. Restivo. Recognizable picture languages. *IJPRAI*, 6(2&3):241–256, 1992.

[21] D. Giammarresi and A. Restivo. Two-dimensional finite state recognizability. *Fundam. Inform.*, 25(3):399–422, 1996.

[22] D. Giammarresi and A. Restivo. *Two-dimensional languages*, volume 3- Beyond words of *Handbook of Theoretical Computer Science*, chapter 4, pages 215–267. Springer-Verlag New York, 1997.

[23] D. Giammarresi, A. Restivo, S. Seibert, and W. Thomas. Monadic second-order logic over rectangular pictures and recognizability by tiling systems. *Information and Computation*, 125(1):32 – 45, 1996.

[24] Dora Giammarresi. Computing languages by (bounded) local sets. In Zoltán Ésik and Zoltán Fülöp, editors, *Developments in Language Theory*, volume 2710 of *Lecture Notes in Computer Science*, pages 304–315. Springer, 2003.

[25] E. Grädel. Capturing complexity classes by fragments of second order logic. *Theoretical Computer Science*, 101:35–57, 1992. A preliminary version appeared in Proceedings of 6th IEEE Conference on Structure in Complexity Theory, Chicago 1991, 341–352.

[26] E. Grädel, Ph. G. Kolaitis, L. Libkin, M. Marx, J. Spencer, M. Y. Vardi, Y. Venema, and S.Weinstein. *Finite Model Theory and Its Applications*. Texts in Theoretical Computer Science. Springer, 2007.

[27] E. Grandjean. First-order spectra with one variable. *Journal of Computer and System Sciences*, 40:136–153, 1990.

[28] E. Grandjean and F. Olive. Graph properties checkable in linear time in the number of vertices. *Journal of Computer and System Sciences*, 68:546–597, 2004.

[29] E. Grandjean and F. Olive. Descriptive complexity of picture languages. In *Proc. Annual Conference of the EACSL (CSL'12), LNCS*, 2012.

[30] E. Grandjean and T. Schwentick. Machine-independent characterizations and complete problems for deterministic linear time. *SIAM Journal on Computing*, 32(1):196–230, 2002.

[31] M. Grohe and T. Schwentick. Locality of order-invariant first-order formulas. *ACM Trans. Comput. Log.*, 1(1):112–130, 2000.

[32] W. Hanf. Model-theoretic methods in the study of elementary logic. In J. Addison, L. Henkin, and A. Tarski, editors, *The Theory of Models*, pages 132–145. North Holland, 1965.

[33] L. Hella, L. Libkin, and J. Nurmonen. Notions of locality and their logical characterizations over finite models. *J. Symb. Log.*, 64(4):1751–1773, 1999.

[34] N. Immerman. *Descriptive complexity*. Graduate texts in computer science. Springer, 1999.

[35] C. Iwamoto, H. Yoneda, K. Morita, and K. Imai. A time hierarchy theorem for nondeterministic cellular automata. In Jin yi Cai, S. Barry Cooper, and Hong Zhu, editors, *TAMC*, volume 4484 of *Lecture Notes in Computer Science*, pages 511–520. Springer, 2007.

[36] A.S. Kahr, E.F. Moore, and H. Wang. Entscheidungsproblem reduced to the forall-exists-forall case. *Proc. Nat. Acad. Sci. U.S.A*, vol.48:pp.365–377, 1962.

[37] J. Kari. *Basic Concepts of Cellular Automata*, pages 3–24. Volume 1 of Rozenberg et al. [58], 2012.

[38] M. Kutrib. Nondeterministic cellular automata and languages. *Int. J. General Systems*, 41(6):555–568, 2012.

[39] M. Latteux and D. Simplot. Recognizable picture languages and domino tiling. Internal Report IT-94-264, Laboratoire d'Informatique Fondamentale de Lille, Université de Lille, France, 1994.

[40] M. Latteux and D. Simplot. Context-sensitive string languages and recognizable picture languages. *Inf. Comput.*, 138(2):160–169, 1997.

[41] M. Latteux and D. Simplot. Recognizable picture languages and domino tiling. *Theor. Comput. Sci.*, 178(1-2):275–283, 1997.

[42] C. Lautemann, N. Schweikardt, and T. Schwentick. A logical characterisation of linear time on nondeterministic turing machines. In *Proc. 14th Symposium on Theoritical Aspect of Computer Science (STACS'99)*, pages 143–152, 1999.

[43] L. Libkin. Logics with counting and local properties. *ACM Trans. Comput. Log.*, 1(1):33–59, 2000.

[44] L. Libkin. Logics capturing local properties. *ACM Trans. Comput. Log.*, 2(1):135–153, 2001.

[45] L. Libkin. *Elements of Finite Model Theory*. Springer, 2004.

[46] Steven Lindell. A normal form for first-order logic over doubly-linked data structures. *Int. J. Found. Comput. Sci.*, 19(1):205–217, 2008.

[47] K. Lindgren, C. Moore, and M. Nordahl. Complexity of two-dimensional patterns. *Journal of Statistical Physics*, 91(5-6):909–951, 1998.

[48] O. Matz and N. Schweikardt. Expressive power of monadic logics on words, trees, pictures, and graphs. In Flum et al. [18], pages 531–552.

[49] O. Matz, N. Schweikardt, and W. Thomas. The monadic quantifier alternation hierarchy over grids and graphs. *Inf. Comput.*, 179(2):356–383, 2002.

[50] Oliver Matz. One quantifier will do in existential monadic second-order logic over pictures. In *MFCS*, pages 751–759, 1998.

[51] Oliver Matz and Wolfgang Thomas. The monadic quantifier alternation hierarchy over graphs is infinite. In *LICS*, pages 236–244, 1997.

[52] R. McNaughton and S. Papert. *Counter-free automata*. M.I.T. Press Cambridge, Mass., 1971.

[53] Martin Otto. A note on the number of monadic quantifiers in monadic Σ^1_1. *Inf. Process. Lett.*, 53(6):337–339, 1995.

[54] Y. Ozhigov. Computations on nondeterministic cellular automata. *Inf. Comput.*, 148(2):181–201, 1999.

[55] Victor Poupet. Cellular automata: Real-time equivalence between one-dimensional neighborhoods. In *STACS*, pages 133–144, 2005.

[56] K. Reinhardt. On some recognizable picture-languages. In *MFCS*, pages 760–770, 1998.

[57] K. Reinhardt. The $\#a = \#b$ pictures are recognizable. In *STACS*, pages 527–538, 2001.

[58] G. Rozenberg, T. Bäck, and J.N Kok, editors. *Handbook of Natural Computing*. Springer, 2012.

[59] A. Schönhage. Storage modification machines. *SIAM J. Comput.*, 9(3):490–508, 1980.

[60] N. Schweikardt. The monadic quantifier alternation hierarchy over grids and pictures. In *CSL*, pages 441–460, 1997.

[61] T. Schwentick. On winning ehrenfeucht games and monadic np. *Ann. Pure Appl. Logic*, 79(1):61–92, 1996.

[62] T. Schwentick and K. Barthelmann. Local normal forms for first-order logic with applications to games and automata. In Michel Morvan, Christoph Meinel, and Daniel Krob, editors, *STACS*, volume 1373 of *Lecture Notes in Computer Science*, pages 444–454. Springer, 1998.

[63] Detlef Seese. Linear time computable problems and logical descriptions. *Electr. Notes Theor. Comput. Sci.*, 2:246–259, 1995.

[64] Detlef Seese. Linear time computable problems and first-order descriptions. *Math. Struc. in Computer Science*, 6:505–526, 1996.

[65] J.I. Seiferas, M.J. Fischer, and A.R. Meyer. Separating nondeterministic time complexity classes. *J. ACM*, 25(1):146–167, 1978.

[66] R. Sommerhalder and S. C. van Westrhenen. Parallel language recognition in constant time by cellular automata. *Acta Inf.*, 19:397–407, 1983.

[67] V. Terrier. *Language Recognition by Cellular Automata*, pages 123–158. Volume 1 of Rozenberg et al. [58], 2012.

[68] J.W. Thatcher and J. B. Wright. Generalized finite automata theory with an application to a decision problem of Second-Order logic. *Mathematical Systems Theory*, 2(1):57–81, March 1968.

[69] W. Thomas. On logics, tilings, and automata. In Javier Leach Albert, Burkhard Monien, and Mario Rodríguez-Artalejo, editors, *ICALP*, volume 510 of *Lecture Notes in Computer Science*, pages 441–454. Springer, 1991.

[70] Wolfgang Thomas. Classifying regular events in symbolic logic. *J. Comput. Syst. Sci.*, 25(3):360–376, 1982.

[71] B.A. Trakhtenbrot. Finite automata and the logic of one-place predicates. *Siberian Math. J.*, 3:103–131, 1962. English translation in: AMS Transl. 59 (1966) 23–55.

[72] P. van Emde Boas. Dominos are forever. *Proc. 1st GTI Workshop, UGH, Paderborn*, pages 75–95, 1982.

[73] P. van Emde Boas. The convenience of tilings. *Complexity, Logic and Recursion Theory, Lecture Notes in Pure and Applied Mathematics*, 187:331–363, 1997.

[74] H. Wang. Dominoes and the aea case of the decision problem. In *Proceedings of the Symposium on the Mathematical Theory of Automata, April 1962, New York, Brooklyn: Polytechnic Press*, pages pp. 23–55., 1963.

[75] H. Wang. *Computation, Logic, Philosophy. A Collection of Essays*. Beijing: Science Press. Dordrecht: Kluwer Academic Publishers, 1990.